Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS

DB2 Performance Monitoring

Ken Kelley

Senior Technical Sales Specialist

IBM Software Group

September 11, 2008

ON DEMAND BUSINESS™

Disclaimer

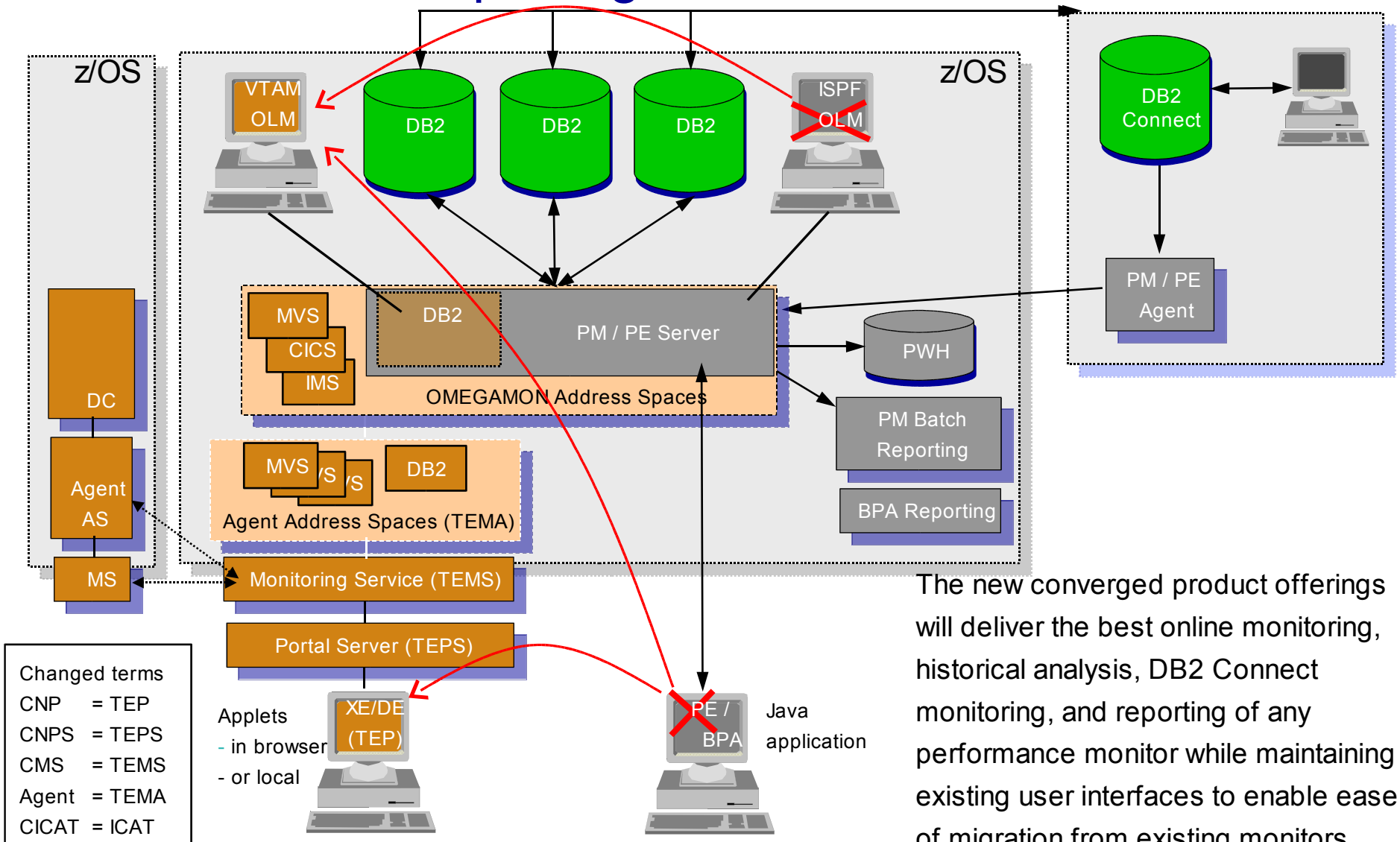
- The information contained in this presentation has not been submitted to any formal IBM review and is distributed on an "As Is" basis without any warranty either expressed or implied.

- The information is subject to change with new versions and enhancements to the software.

Agenda

- OMPE Overview
- Trace Data
- Tuning Methodology
- OMPE 3270 Classic Interface
- OMPE Batch Reports

One Server / Reporting / TEP and UI Consolidation



The new converged product offerings will deliver the best online monitoring, historical analysis, DB2 Connect monitoring, and reporting of any performance monitor while maintaining existing user interfaces to enable ease of migration from existing monitors.

Features and Components

- Real Time Thread Analysis
  - ✓Thread performance
  - ✓Thread Detail
  - ✓Triggers, Procedures, & UDFs
- Real Time - DB2 subsystem
  - ✓Virtual & EDM Pool analysis
    - ✓Pool performance & snapshot analysis
  - ✓Locking & Logging Analysis
  - ✓Storage Analysis
- Application Trace Facility
  - ✓Detailed performance tracing
- Choice Of Interfaces
  - ✓(TEP, PE GUI, 3270)
- Buffer Pool Analysis (PE only)
- DB2 Connect Monitoring
- Object Analysis
  - ✓I/O & get page analysis
  - ✓Correlate by object & App
- Locking & Lock Conflicts
- Historical Analysis
  - ✓Near-term history online
  - ✓Snapshot History
  - ✓Batch reporting
  - ✓Performance Warehouse
  - ✓XE Tivoli Warehouse
- DB2Plex Monitoring View
  - ✓CF structure & lock analysis
- Automation capabilities
- zIIP Engine utilization

Real-Time Monitoring

- Real-time online monitoring of multiple DB2 subsystems:
  - System statistics
  - Thread summary and detail
  - System configuration parameters (DSNZPARM)
  - Bottlenecks (locking conflicts)
  - Object analysis
  - Periodic and Event exception processing
  - Application Trace Facility and SQL Trace
  - Historical data (near-term, snapshot, etc.)
  - Graphical view of important performance data
  - DB2 Connect monitoring

Trace Data

- Event Trace Data
  - Used by online and batch reporting
  - Trace records can be sent to multiple destinations
  - Allows problem analysis at a very detailed level
  - Can be aggregated for trend analysis
  - Load into the DB2 PE Performance database
  - Application Trace Facility
  - SQL Trace
- Snapshot Data
  - Used by online monitoring only
  - Takes a snapshot of current DB2 and application activities
  - Performs continuous monitoring of individual applications and immediately reports events such as conflicts and exceptions
  - Useful when a problem has recently occurred or persists

History

- Recent History
  - Short range history (minutes, hours, days)
  - Near-term / Short-term / Snapshot
  - Collects or summarizes information from IFI
  - Interval and amount / type of data are customizable
  - Uses a history dataset with a wrap-around mechanism
  - Online user interfaces only
- Long-term History
  - Long range history (days, weeks, months)
  - Usually SMF records
  - Batch Reporting and online interfaces
  - Performance Warehouse

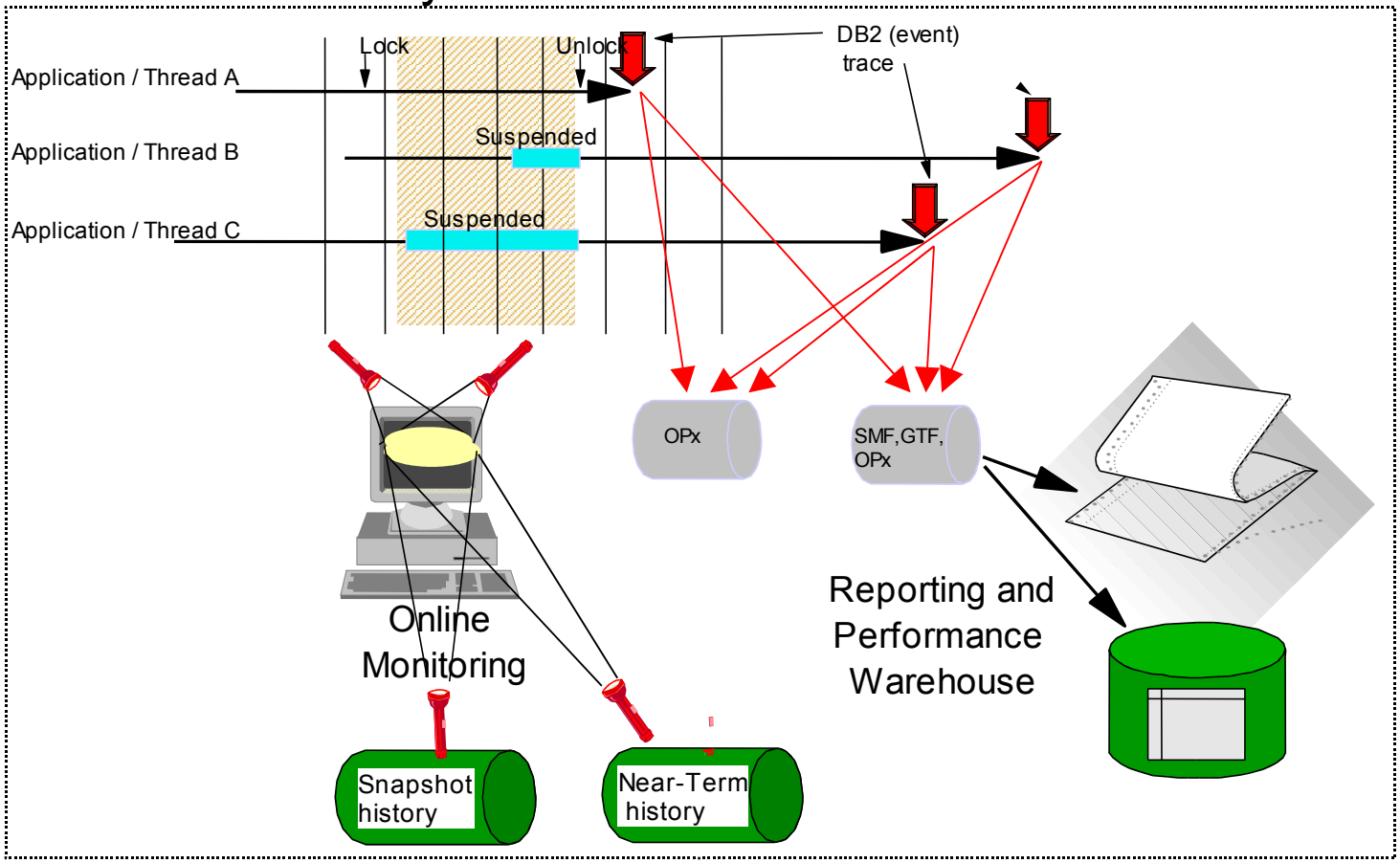
History

- Near-term history- Online monitor (3270 classic)
- Long-term history- Reports, Performance DB
- Snapshot history- Online Monitor (PE,ISPF)
- Short-Term history- TEP GUI

Based on DB2 (event) traces

Based on DB2 snapshot data

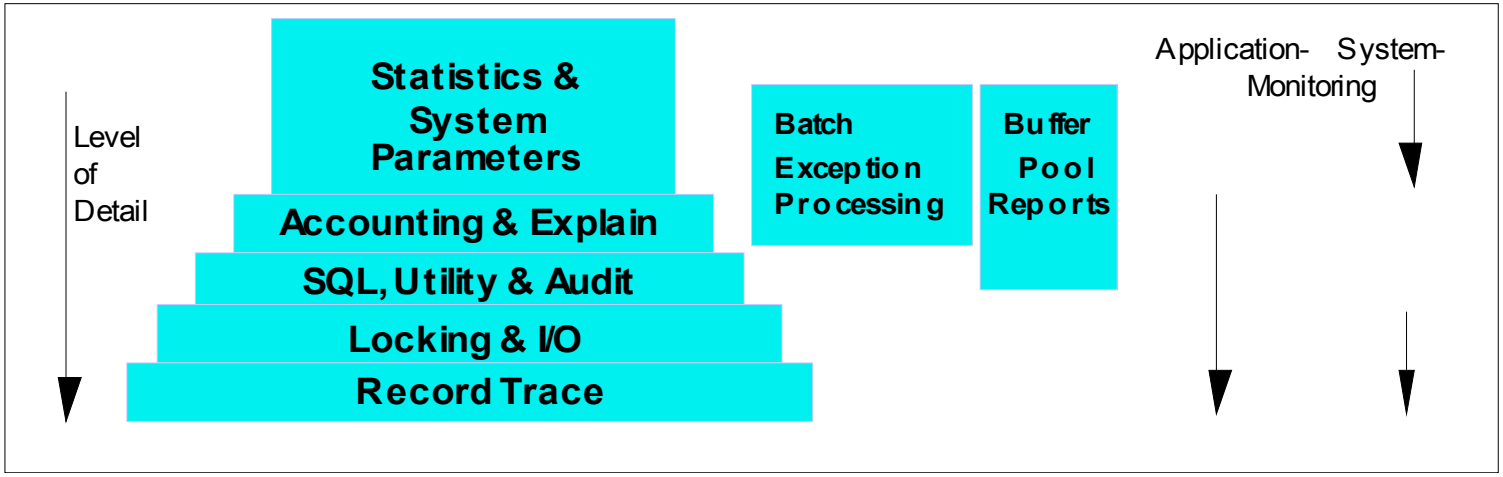
Collected and saved by TEMS or TEMA



Batch Reporting

Report facility that

- Takes SMF, GTF or TSO data sets (collected by DB2 Performance Expert Collect Report Data) as input
- Generates a variety of customizable reports and traces:



Invocation

- Online or via MVS JCL (**I**nteractive **R**eport **F**acility or manually)
- or via workstation GUI (Statistics & Accounting Report)
  - Result shown in browser window
- Integrated into Online monitoring (SQL activity tracing)

Reduction of trace information for loading into the Performance DB

Performance Warehouse

Functions

- An fully automated warehouse for accounting and statistic information
- Can contain raw and summarized data from multiple DB2 subsystems
- Performance Database is built, controlled and maintained by OM Server
- PWH Client (PC based GUI) allows the collection of data from DB2, define ETL process and generate Reports for it
- Tasks can be scheduled or executed on demand
- Predefined SQL Queries allow to evaluate performance data
- Rules-of-thumb check for potential problems or tuning ways

Performance Warehouse Client

- Java front-end for PWH
- Running under Windows
- Integrated into Performance Expert Client
- Requires DB2 Connect

Input Data

- Pre-processed data from Report Facility
- OPx data collected via CRD
- GTF data
- SMF data

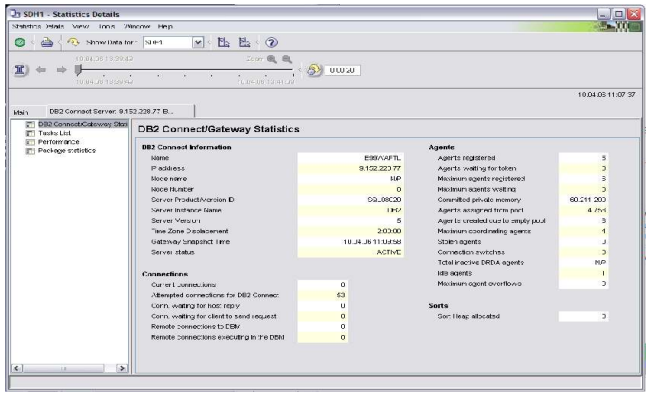
DB2 Connect Monitoring

Functions

- Allows to monitor DB2 Connect gateways connected to a DB2 on z/OS or LUW (end-to-end monitoring)
- Fully integrated into Classic Interface, Tivoli Enterprise Portal and PE Client
- Shows DB2 Connect activity either from the perspective of the gateway or from the DB2 host
- Information about DB2 threads on the host are fully correlated with information about the thread on the DB2 Connect gateway
- Performance between gateway and host is regularly measured to inform immediately about problems
- Allows tuning of DB2 Connect gateway but also problem determination

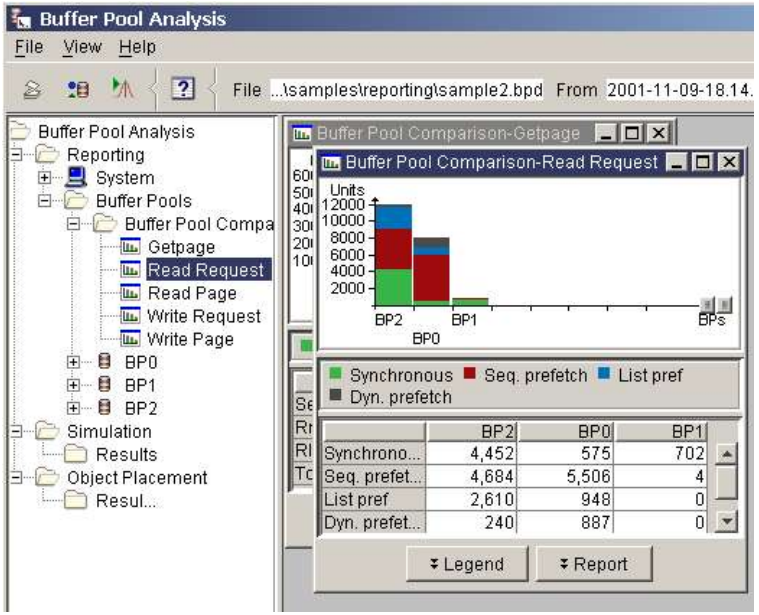
Show Information

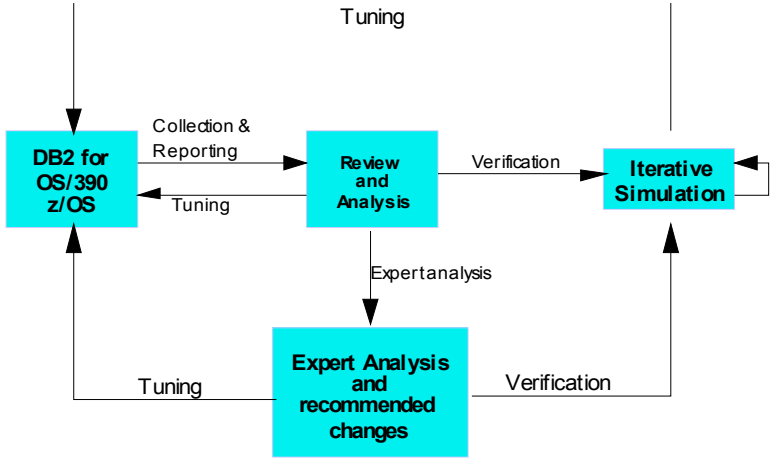
- Network between DB2 client and gateway
- Network between DB2 gateway and host
- System utilization of gateway machine
- DB2 Connect configuration
- DB2 Connect utilization
- Connected DB2 Clients
- Performance of connection to DB2 host



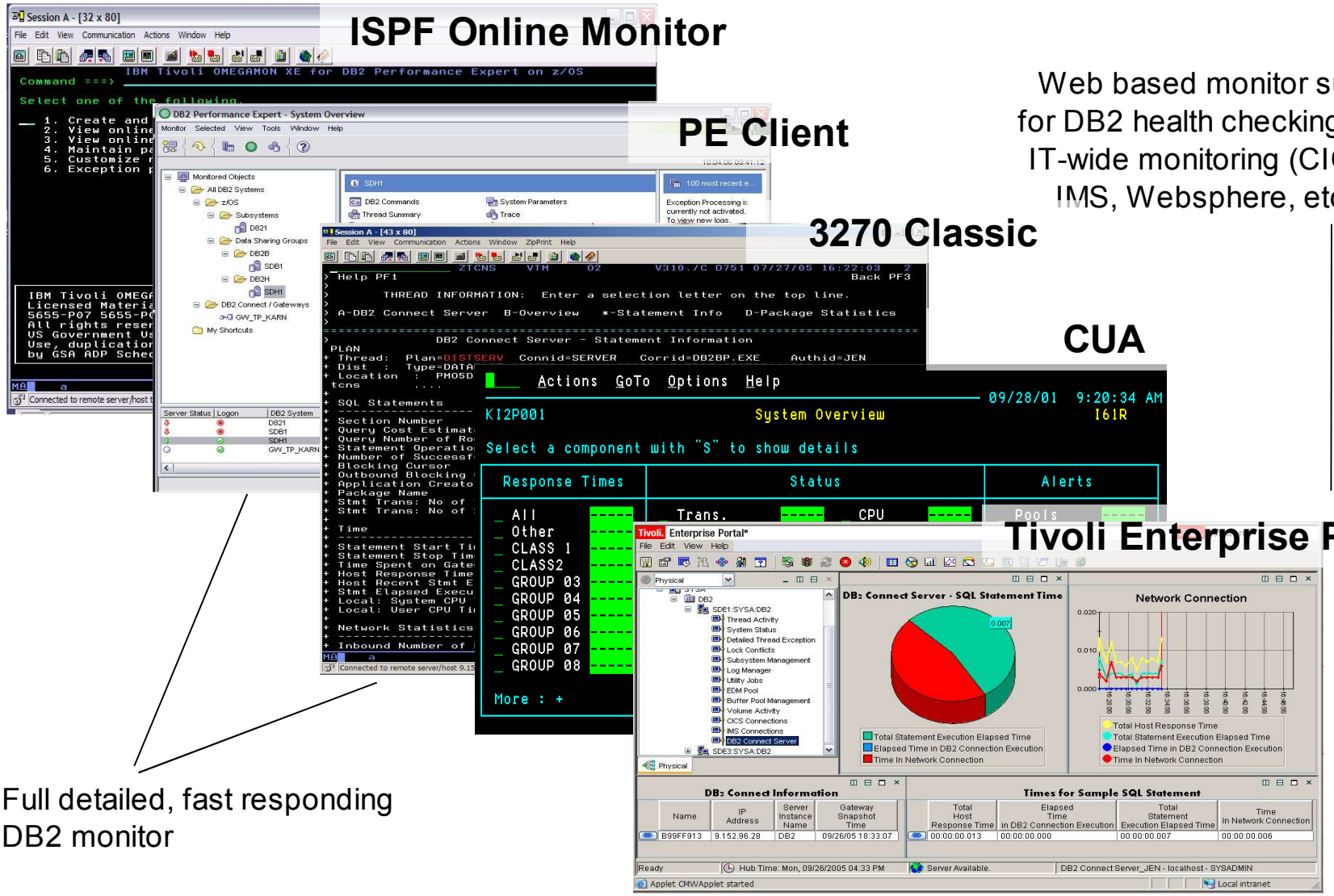
Buffer Pool Analysis

- **Collects** buffer pool data
  - as summary or detailed data
  - continuously or in sampling mode
  - in Online and Batch
- Generates various **reports and displays** results in multiple formats for BP and GBP (including graphical end-user interface)
- Provides expert knowledge and recommendations
- Recommends **object placements, BP size & thresholds**
- **Generates ALTER statements** for the recommendation
- Provides **simulation** for planned changes
- Makes it easy to tune your buffer pools

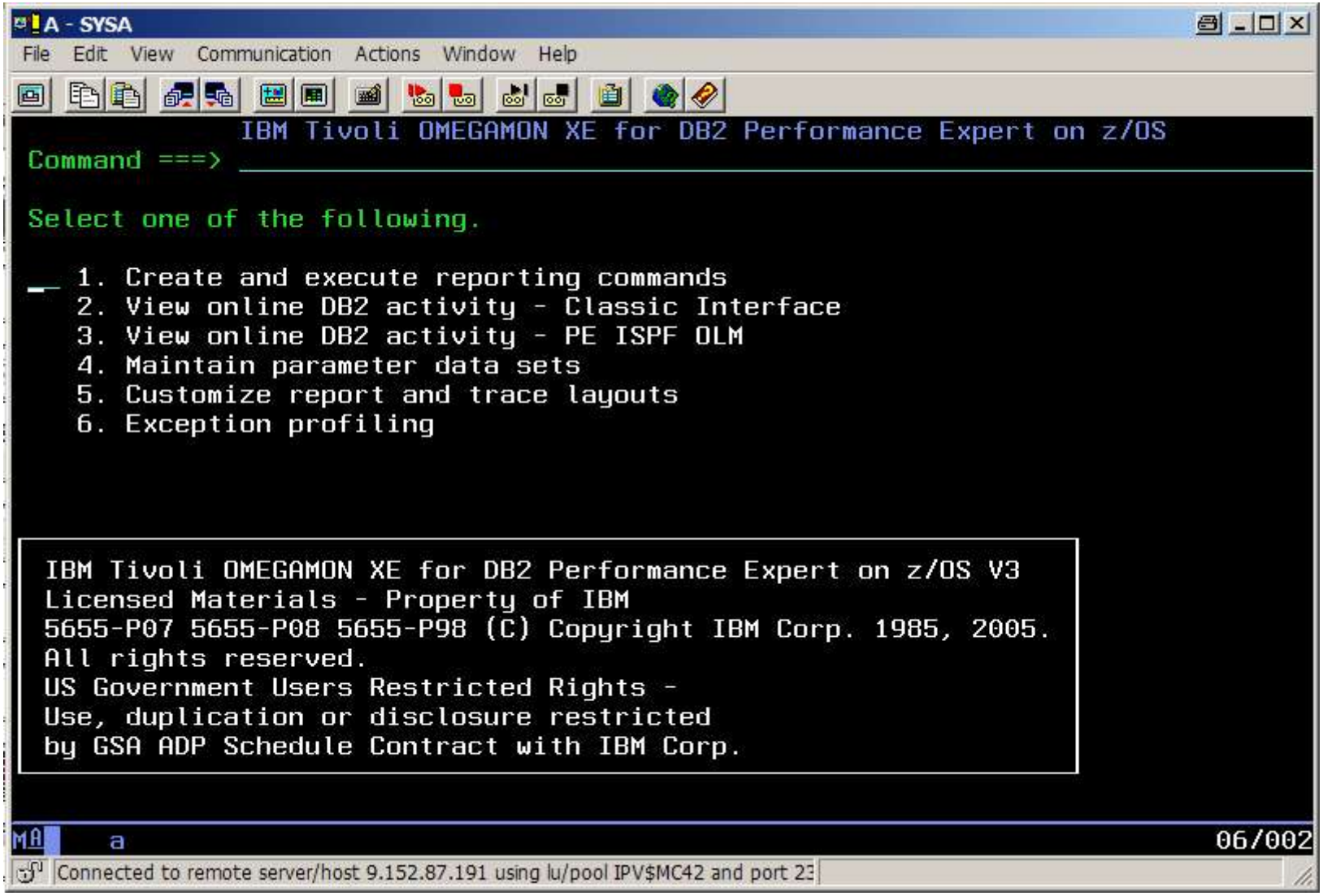




User Interfaces



ISPF Dialog

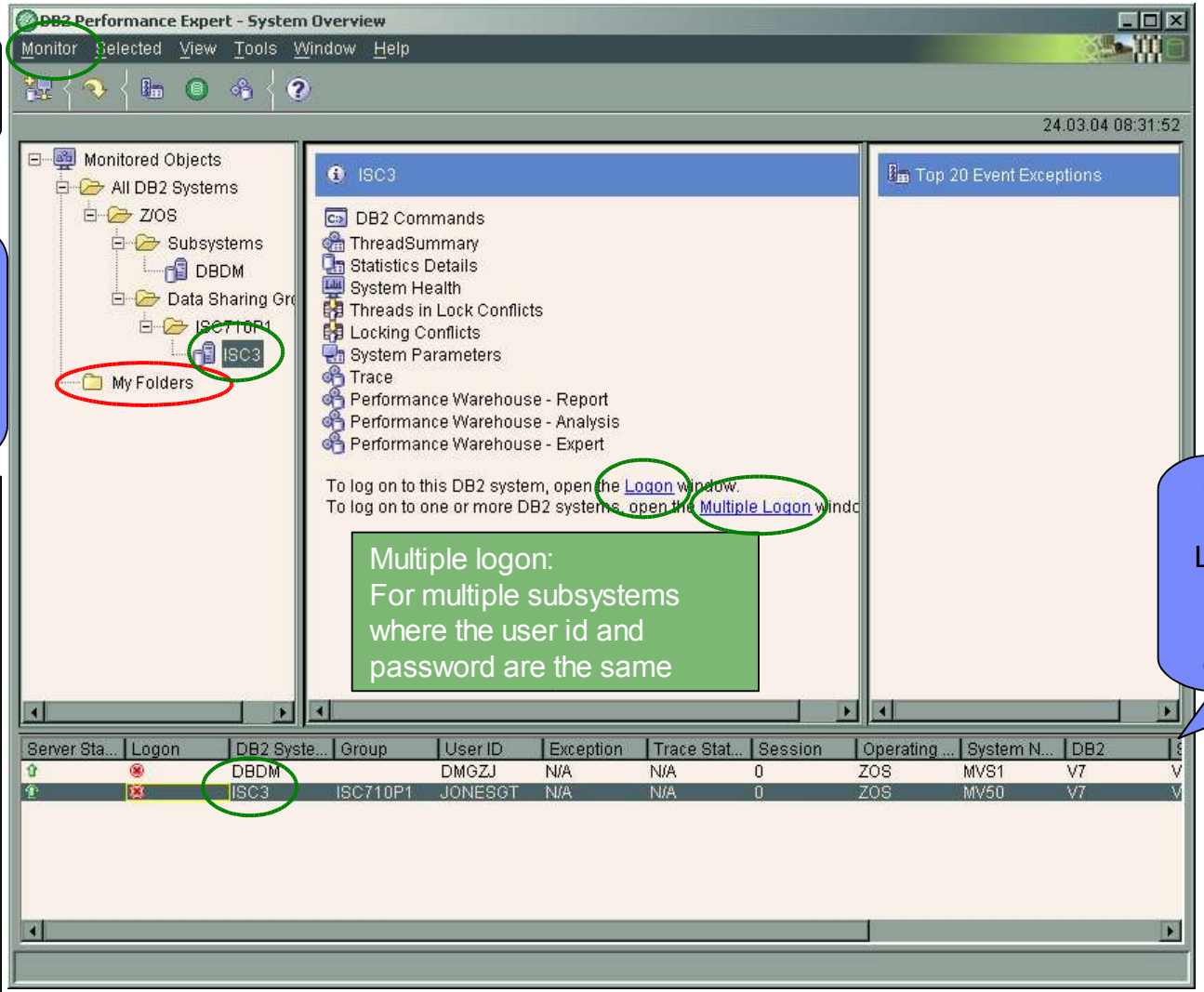


PE Client

Monitored Objects pane

Group DB2 subsystems in user-defined views under 'My Folders'

Logging on:
- From Server Status pane
- From Monitored Objects pane
- From Selected Subsystem pane
- From menu bar
- Indicated by green ring



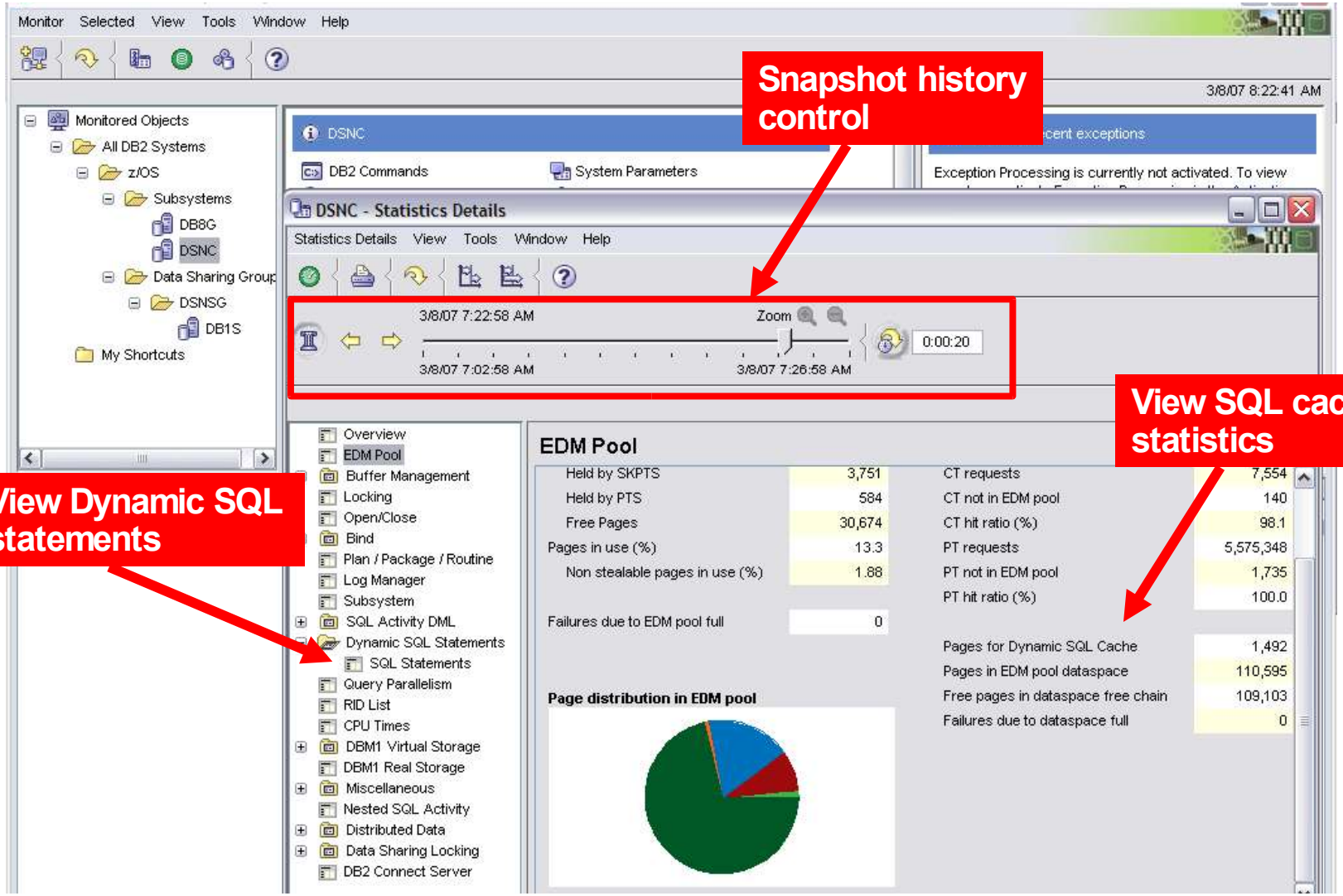
Multiple logon:
For multiple subsystems where the user id and password are the same

Exception Event pane

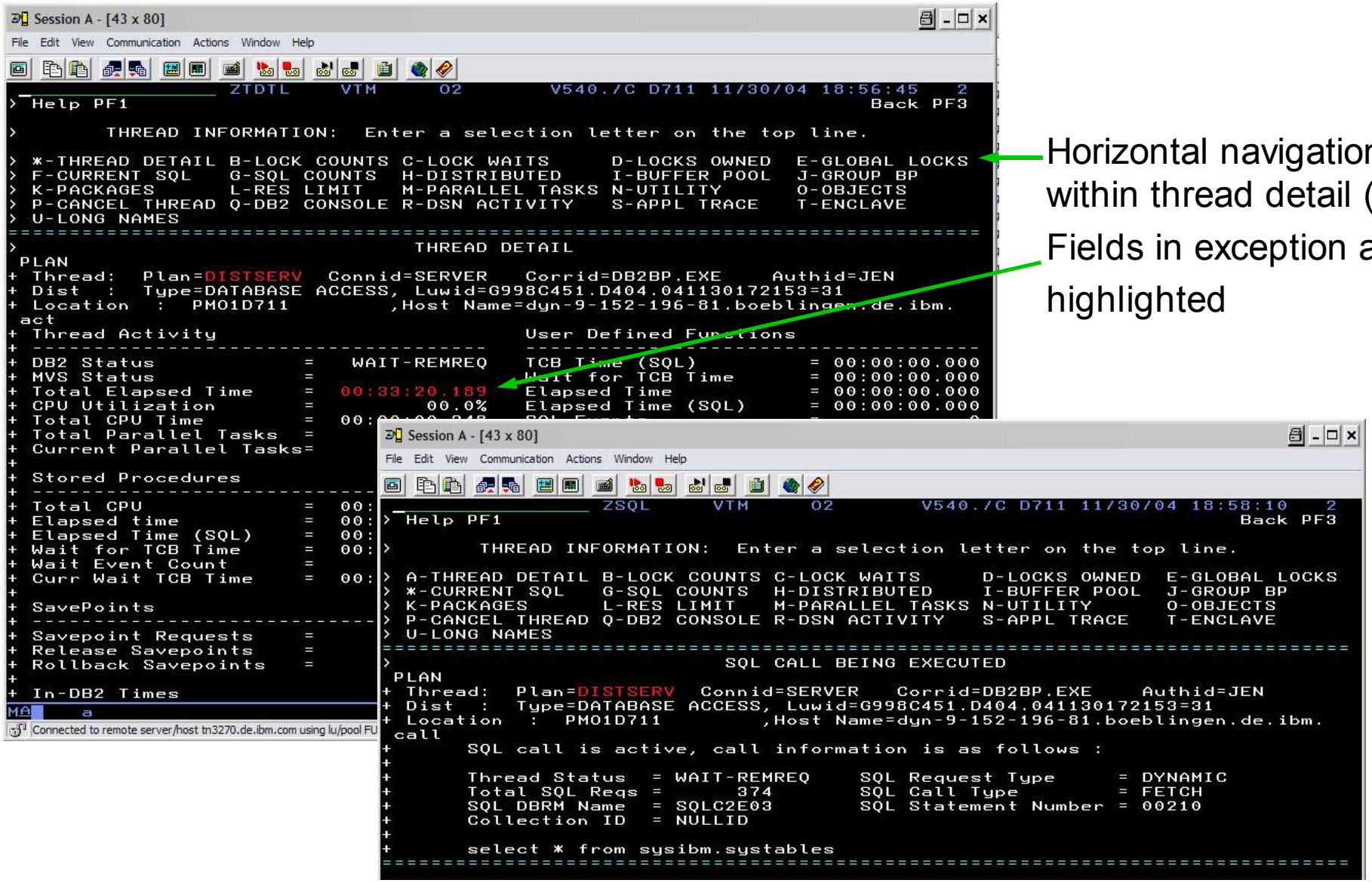
Selected Subsystem pane.
Launch PE tasks from here by double-clicking,
or from the menu bar

PE Server Status pane

PE Client - Real-time Data and Snapshot History



3270 Classic - Thread Snapshot



Horizontal navigation within thread detail (*)

Fields in exception are highlighted

Tivoli Enterprise Portal (Web Browser)

Easy to use Browser controls

Plug and Play components

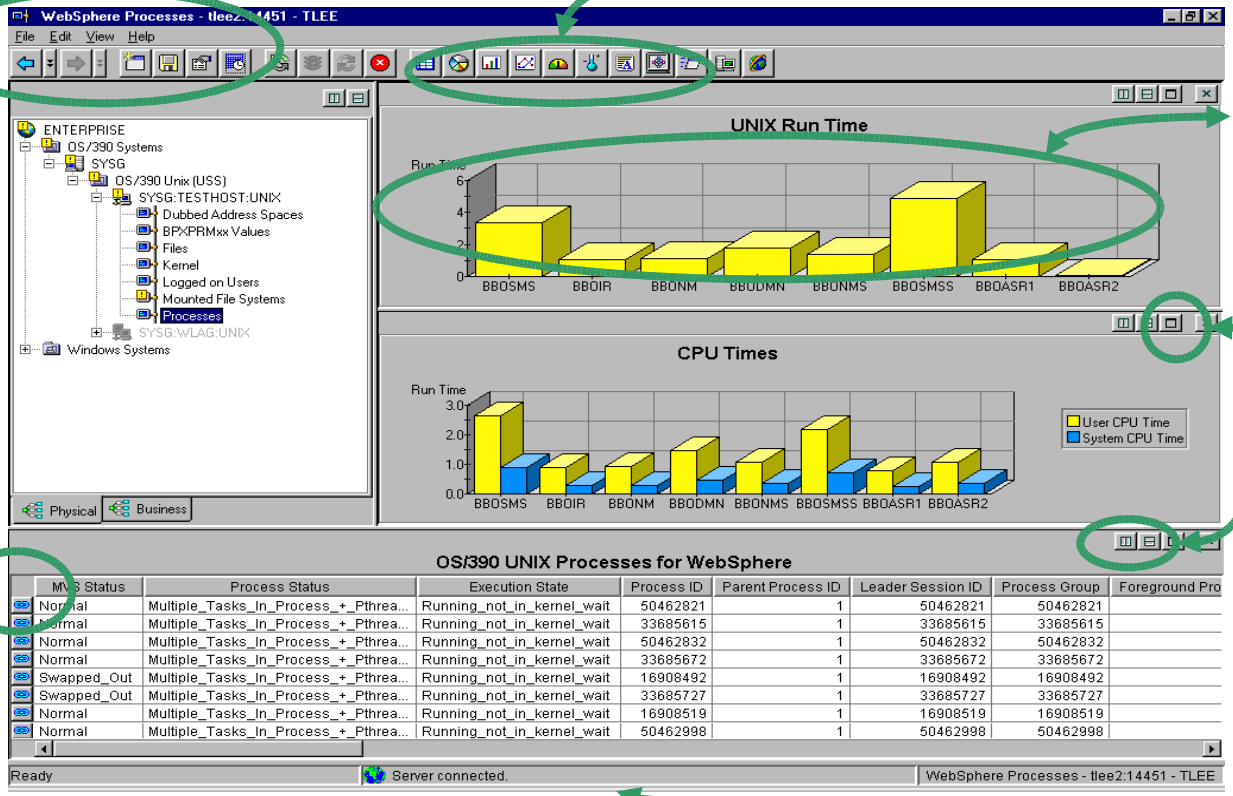
Personalized Views

View Zoom

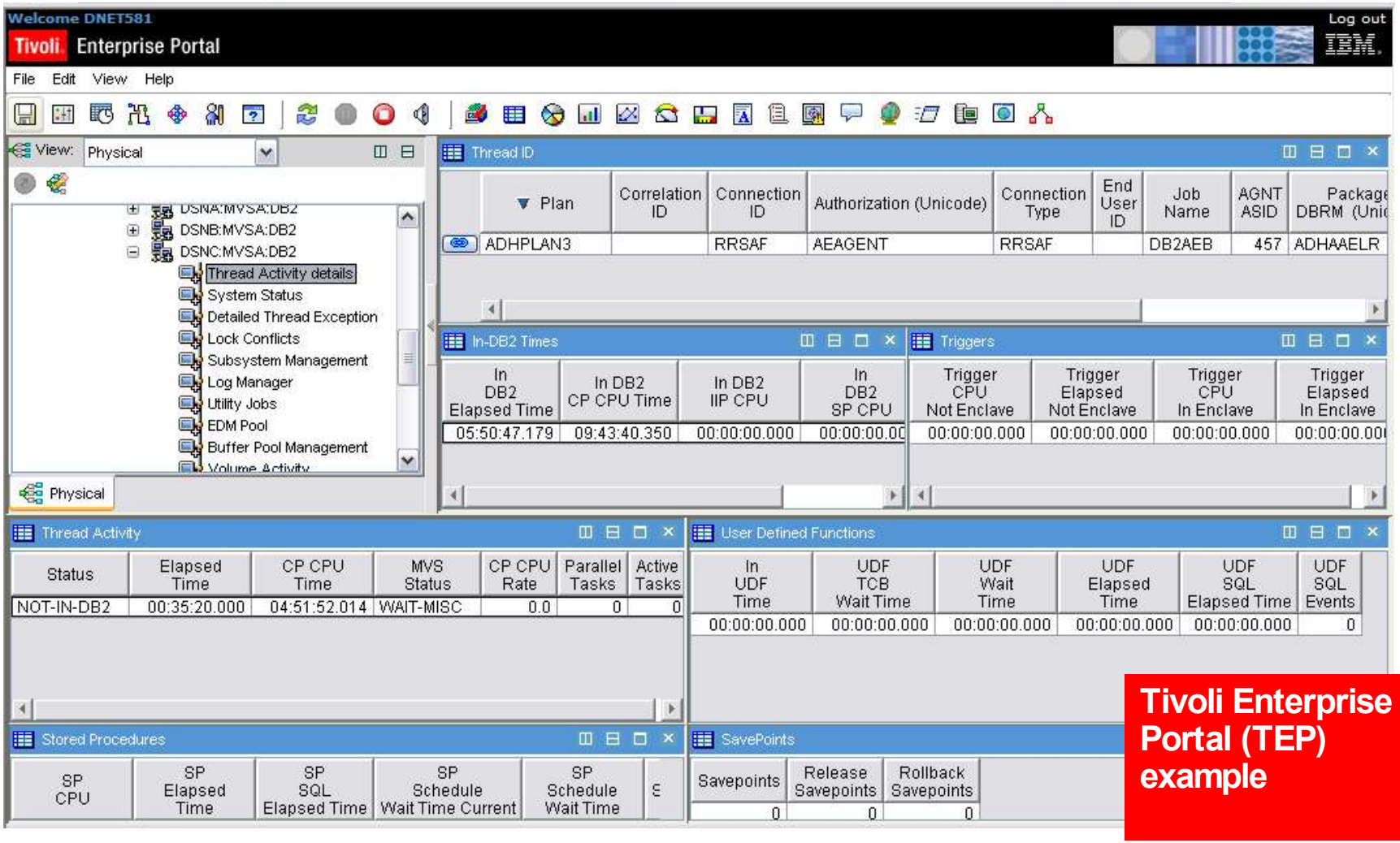
Splitter controls

Intelligent Linking

Persistent customized workspaces



TEP - Thread Detail



Tivoli Enterprise Portal (TEP) example

Trace Data

IBM Software Group

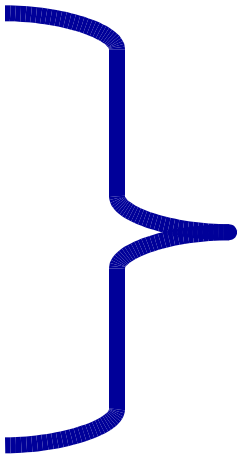
ON DEMAND BUSINESS™

Trace Data Topics

- Trace Data Fundamentals
- Statistics Traces
- Accounting Traces
- Performance Traces

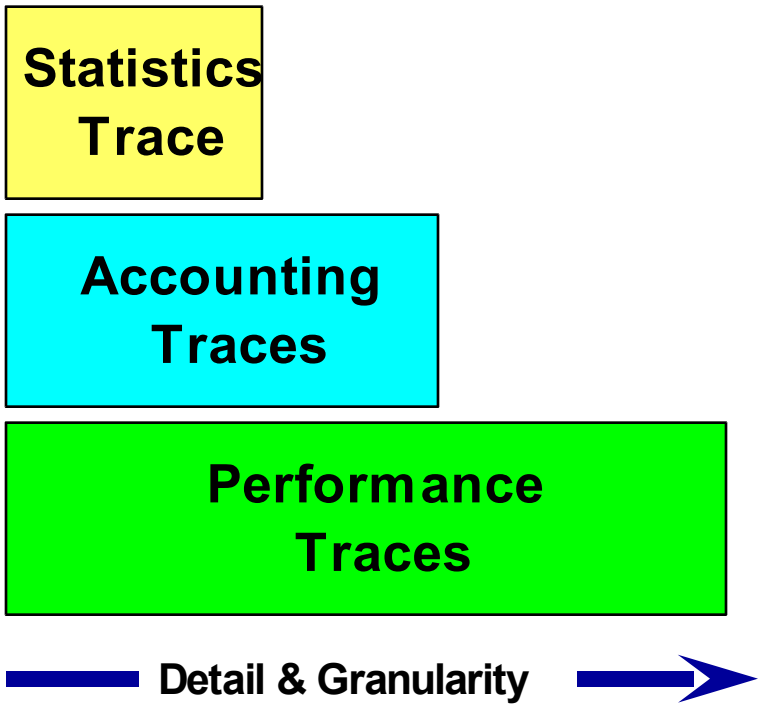
Types of Trace Data

- Statistics
  - DB2 subsystem-wide DB2 measurements
- Accounting
  - Application execution measurements
- Performance
  - Measurements of specific events
- Monitor
  - Trace data for program access (IFI interface)
- Audit
  - Record security and access events
- Global
  - IBM support use



Use these
For performance
analysis

Trace Data Granularity



Subsystem level
# of calls and type by time interval
Identify high level issues

Application level
# of calls, type, duration by type, wait time
Identify problem applications

Detail event level
Call level detail, duration, events
Isolate call level problems

- Corresponding trace types gather varying levels of detail and information
- More detailed traces may (most likely will) result in additional trace overhead

Trace Overhead

- DB2 Accounting Class 1 and 3 and Statistics Traces
  - Total cost of about 2 percent to 5
- Accounting Class 2 may add 2 to 5 percent
  - Dependent upon DB2 application
- Audit Trace
  - Typically less than 5 percent
- Performance Trace
  - Very Dependent upon trace classes and IFCIDs started
  - Overhead anywhere from 20 percent to 100 percent

Trace Architecture

- Each trace record type has a specified format and ID (the IFCID)
  - IFCID = Instrumentation Facility ID
- Traces may be activated
  - Automatically at DB2 startup
  - Manually by the DB2 START TRACE command
- Trace destinations
  - SMF- History facility for all subsystems on z/OS
  - GTF - Generalized Trace Facility
  - OPx - Online destination used by monitoring tools
  - SRV - Serviceability

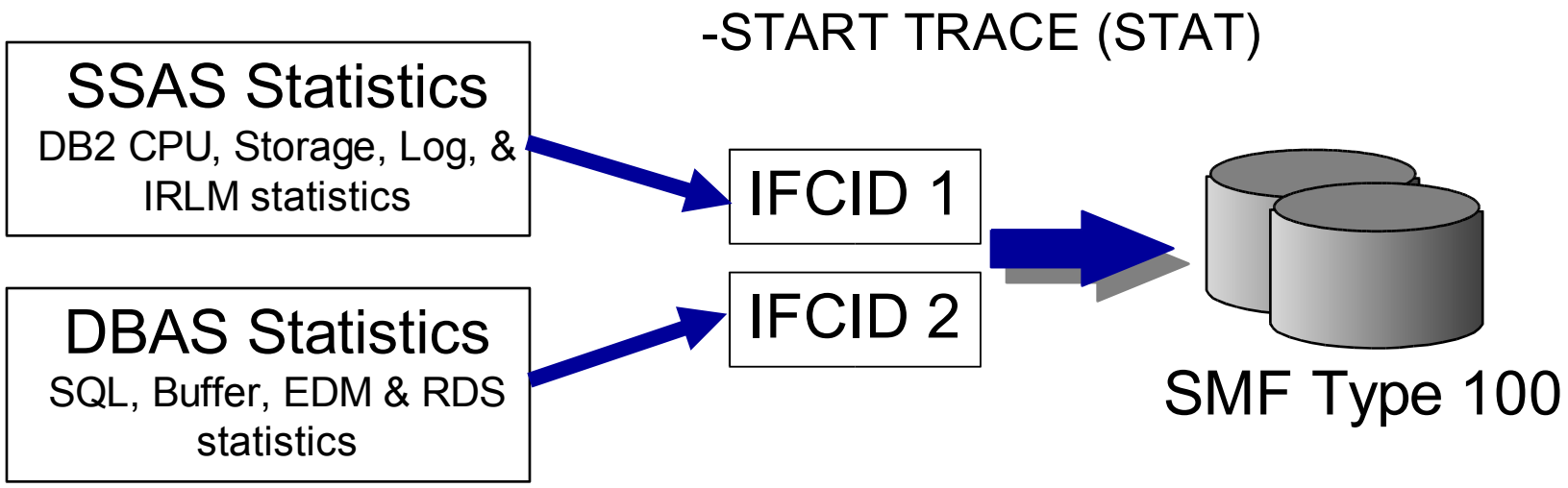
Trace Documentation

- Traces are numbered and well documented
- Library SDSNIVPD member DSNWMSGS contains descriptions of all trace IFCIDs
- Download and print this member out

Example of DSNWMSGS

```
_____________________ IFCID 0001 ________________________________
(RMID 26)
DB2 STATISTICS RECORD.
DB2 WRITES STATISTICS DATA AS SMF TYPE 100 RECORDS
(SUBTYPE 0).
FOR SYSTEM SERVICE STATISTICS AND DATABASE STATISTICS:
IFCID 1 IS FOR SYSTEM SERVICES STATISTICS.
IFCID 2 IS FOR DATABASE STATISTICS.
BOTH ARE WRITTEN ON A REGULAR TIME INTERVAL AS SPECIFIED
WITH INSTALL PARAMETER STATISTICS TIME, PANEL DSNTIPN.
MOST COUNTS IN THESE RECORDS ARE ACCUMULATED
SINCE DB2 WAS LAST STARTED. HENCE, THE NUMBERS
INCLUDE COUNTS PRIOR TO THE REPORT PERIOD COVERED.
VALUES ARE RESET TO ZERO ONLY WHEN DB2 IS STARTED.
FIELDS IN THIS RECORD ARE GROUPED BY SECTIONS AND
ARE PRESENTED IN THE FOLLOWING ORDER:
SECTION QWS00PSO IS MAPPED BY DSNDQWHS, AND OPTIONALLY BY
      DSNDQWHC, DSHDQWHT, DSNDQWHU, DSNDQWHD, DSNDQWHA
SECTION QWS00R1O (ADDRESS SPACE DATA) IS MAPPED BY
      DSNDQWSA.
SECTION QWS00R2O (INSTRUMENTATION DESTINATION DATA) IS
       MAPPED BY DSNDQWSB.
SECTION QWS00R3O (INSTRUMENTATION DATA) IS MAPPED BY
       DSNDQWSC.
SECTION QWS00R4O (SUBSYSTEM SERVICES DATA) IS MAPPED BY
        DSNDQ3ST.
SECTION QWS00R5O (COMMAND DATA) IS MAPPED BY DSNDQ9ST.
SECTION QWS00R6O (IFC CHECK POINT DATA) IS MAPPED BY
        DSNDQWSD.
SECTION QWS00R7O (LATCH MANAGER DATA) IS MAPPED BY
        DSNDQVLS.
SECTION QWS00R8O (AGENT SERVICES DATA) IS MAPPED BY
        DSNDQVAS.
SECTION QWS00R9O (STORAGE MANAGER DATA) IS MAPPED BY
        DSNDQSST.
SECTION QWS00RAO (LOG MANAGER DATA) IS MAPPED BY
```

Statistics Traces



- System-wide performance data gathered at specified intervals (set in DSNZPARM)
- Written to SMF with type 100 header
  - Two IFCIDs written per period
- Overhead small (approximately 1%)

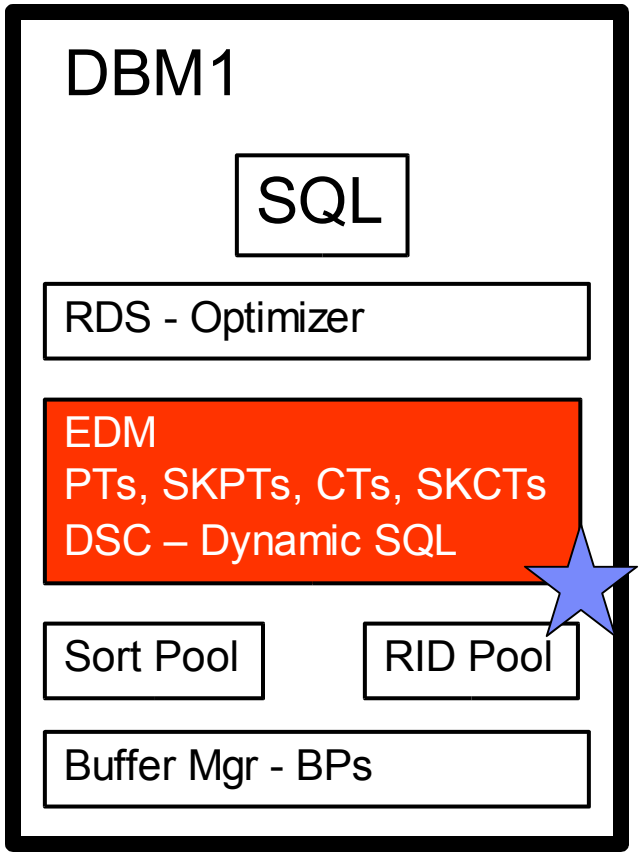
Statistics Counters

- Values can be
  - An accumulated value since the DB2 system was last started
    - For example, the total number of SELECT statements that were executed since the system was last started
  - A current or snapshot value
    - For example, the number of open data sets at the time the DB2 Statistics records pair was externalized
  - A maximum or high water mark value the counter has reached since the time DB2 was last started
    - For example, the maximum number of open data sets at any time since the system was last started.

Using Statistics Data

- Statistics data provides information on the nature and activity of the DB2 workload at the subsystem level
  - How busy is the system
    - Number of getpages, threads or SQL statements
  - What is the nature of the workload
    - Static vs. dynamic SQL
    - Stored Procedure, UDF and Trigger activity
    - Parallelism
- Look for problem indicators and unusual counts
  - Thread queuing
  - Poor pool performance
    - EDM, BP, Sort, RID pools
  - Lock escalation, deadlocks, timeouts
  - High numbers of Aborts

EDM Pool Analysis

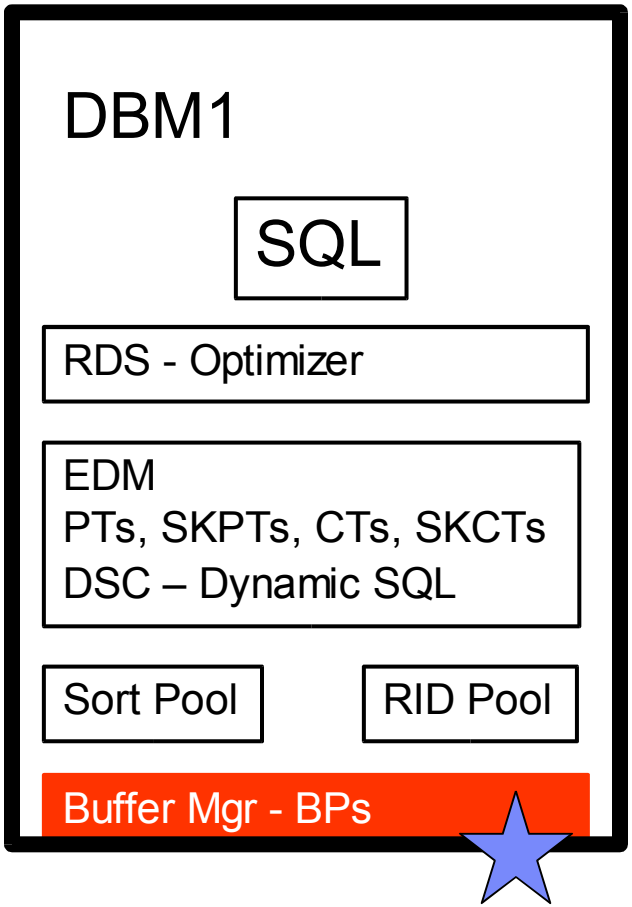


- Is The EDM Pool Sized Appropriately?
  - Pages in EDM pool & percent in use
  - Failures due to pool full
- How Is The EDM Pool Being Used?
  - Page usage counts by type (CT, PT, SKCT, SKPT, DBD, Cache)
- What Is The EDM Pool 'Hit ratio'?
  - Number of requests, Number of requests not found in pool
- Is Dynamic SQL Cache Being used Effectively?
  - Inserts, Requests, Cache size
  - Calculate 'Hit' ratio

EDM Pool Analysis

- Monitor EDM Pool statistics for
  - FAILS DUE TO POOL FULL
  - REQ NOT FOUND IN EDMPOOL
  - PREP_STMT_HIT_RATIO
- Tune EDM pool size
  - Bind option ACQUIRE USE
  - Bind option RELEASE COMMIT for all but most frequently executed plans and packages
  - Reduce DBD size with REORG and MODIFY if many dropped tables in segmented table spaces

Buffer Pool Analysis



- Are Buffer Manager thresholds being reached?
  - Examples - DWTH, DMTH, SPTH
- Is the BP Sized appropriately?
  - Paging for Read or Write
  - Number of currently active buffers
- What Is the BP Hit Ratio?
  - Synchronous Reads, Pages dRead via Prefetch, Getpages

$$\text{Hit Ratio} = \frac{\text{Getpages} - (\text{Sync I/Os} + \text{Pages read via Prefetch})}{\text{Getpages}}$$

Buffer Pool Analysis

- Critical Counters
  - Prefetch Disabled - No Buffer
    - Minimize to zero by increasing BP size
  - Data Manager Critical Threshold
    - Minimize to zero by increasing BP size
  - Page-In for Read / Write
    - Check MVS paging - if short on central storage reduce BP size
  - Synch Reads
    - Minimize if possible by increasing BP size

Lock Performance

- Is There Lock Contention?
  - Timeout & Deadlock counts
  - IRLM Latch Contention
- Is There An Increase In Lock Activity?
  - Lock Requests & Unlock Requests
  - Lock Escalation
  - Are There Data Sharing Lock Activity Issues?
- Lock & Unlock requests
  - IRLM & XES contention
  - False contention

Checkpoint and Log Performance

- Checkpoint activity
  - Number of checkpoints taken
- Potential Logging Bottlenecks?
  - Waits caused by unavailable output buffer
  - Number of log Control Intervals created - calculate rate
- Is There Backout Activity?
  - Reads from Output Buffer
  - Reads from Active Log
  - Reads from Archive Log

Accounting Traces

Accounting Class 1
Thread elapsed times,
SQL stats, Buffer & lock stats

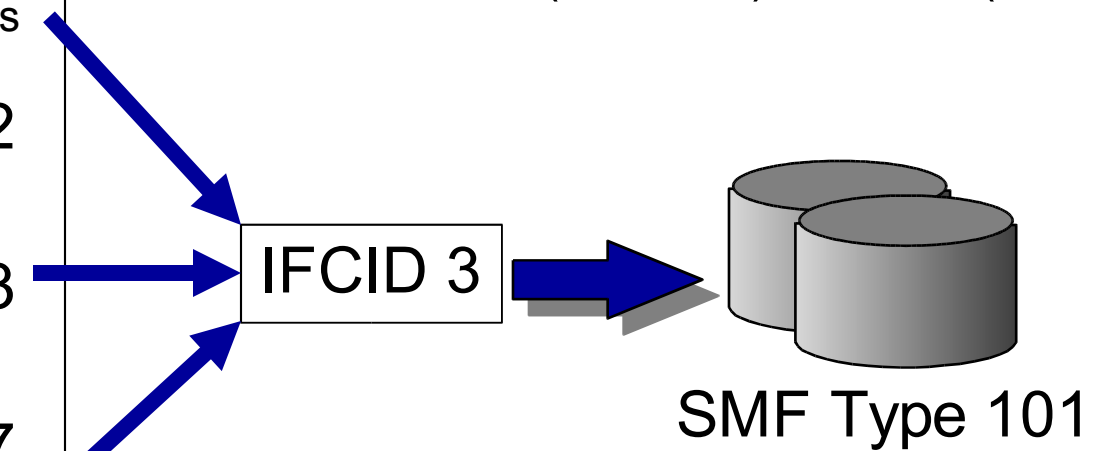
Accounting Class 2
"In-DB2" Time

Accounting Class 3
"Wait" times and counts

Accounting Class 7
"In-DB2" Package level

Accounting Class 8
"Wait" times Package level

-START TRACE (ACCTG) CLASS (1,2,3,7,8)

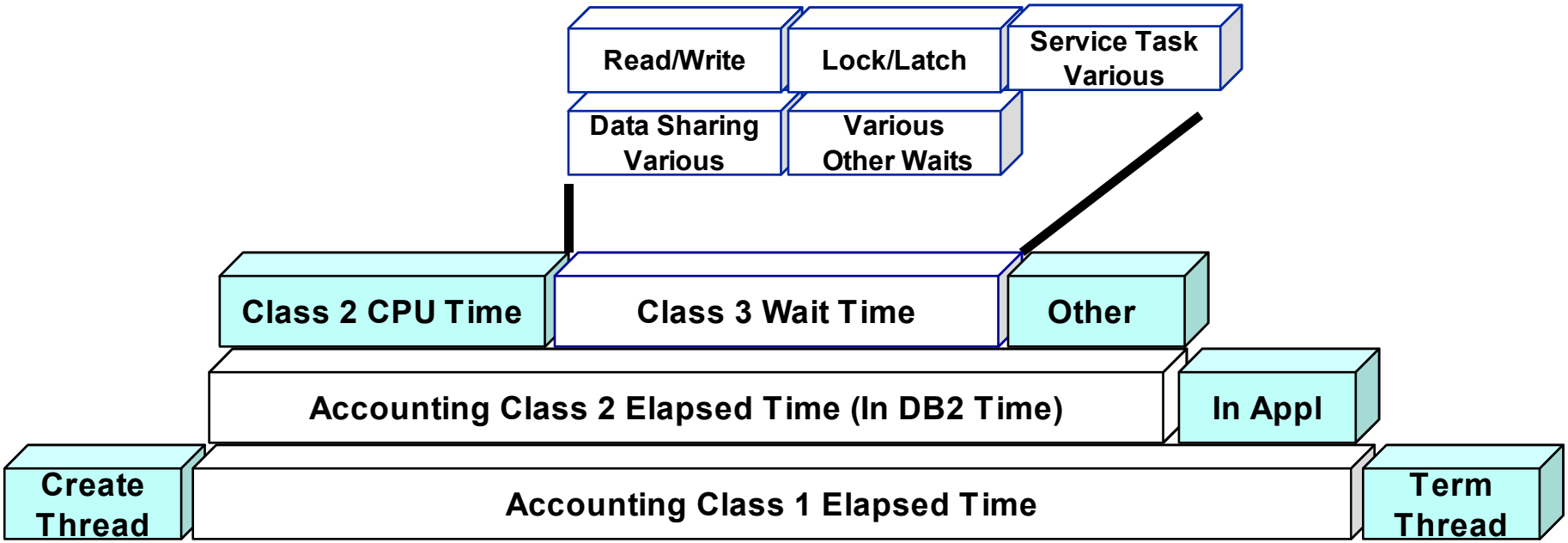


- Data written at thread termination
- Five main classes of accounting traces
- Written to SMF as a type 101 record

Using Accounting Data

- Review accounting data to understand what the applications are doing
  - Number of SQL calls, type of SQL calls, duration of SQL In-DB2 activity, dynamic SQL activity
  - DB2 SQL waits - I/O and lock/latch waits
  - Stored Procedure activity, number of procedure calls and elapsed and In-DB2 times, SP scheduling delays
  - Package level detail
- Accounting traces are the starting point for DB2 application and database performance analysis

Using Accounting Data



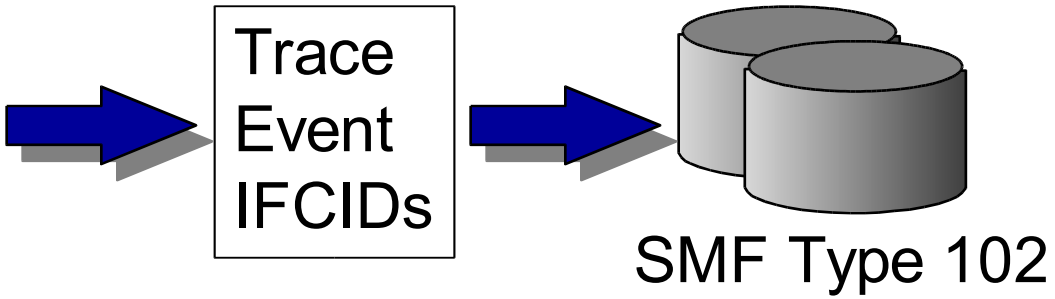
- Accounting traces provide a wealth of information at the application level
  - Elapsed time, CPU time, and wait time for DB2 applications
  - Wait counts and times (I/O, locks, latches, etc)
  - Scan information, Buffer information, SQL counts
- Use application trace data to determine application time line

Performance Traces

Example Trace Classes

Class 1 - Background Events
Class 2 - Subsystem Events
Class 3 - SQL Events
Class 4 - Buffer & EDM
Class 5 - Log Manager
Class 6 - Lock Summary Info
Class 7 - Lock Detail Info
Class 8 - Data Mgr Detail
Class 9 - Sort Detail
Class 10 - Bind Commands Utilities
Class 11 - Dispatching
Class 12 - Storage Mgr
Class 13 - Edit & Validation Exits
Class 14 - In & Out of DB2
Class 15 - Installation Defined
Class 16 - To/from other Locations
Class 17 - Drain/Claim Detail
Class 20 - Data Sharing Summary
Class 21 - Data Sharing Detail
Class 22 - Authorization Exit
Class 23 - Language Env …

-START TRACE (PERFM) CLASS (3)



- Detailed traces of DB2 events
- Each Trace Class consists of multiple event types (IFCIDs)
- Many events may be traced
- Written to SMF as type 102

Using Performance Traces

- Accounting Traces can isolate performance issues down to the plan / dbrm / package level
- Use Performance Traces to isolate further
  - Statement level
  - Detail activity within the statement level
- The most detailed level of tracing activity
  - Performance Trace IFCIDs of events within each class (1 to 334)
  - Many are IFCID pairs (Begin and End event)
- Use judiciously and with caution
  - Understand what is being traced
  - Trace only what is needed to isolate the problem
  - Filter by Plan, Authid, Class, or Location
  - Use the IFCID option to specify IFCIDs in addition to class specified
    - Use Class 30-32 to specify only selected IFCIDs

Performance Trace Overhead

- Low Overhead
  - Class 1 Background Events, Class 2 Subsystem Events, Class 3 SQL Events, Class 10 Utilities & Commands
- Medium Overhead
  - Class 6 Lock Summary, Class 8 Data Mgr scans
- High Overhead
  - Class 4 Buffer Mgr, Class 5 Log Mgr, Class 9 Sort activity
- Very High Overhead
  - Class 7 Lock Detail, Class 13 Edit & Validation

Tuning Methodology

IBM Software Group

ON DEMAND BUSINESS™

Tuning Methodology Topics

- Where Is The Problem?
- DB2 CPU Problems
- DB2 I/O Problems
- Other Wait Problems
- More Details

Tuning Methodology

- Establish Service Level Agreements
- Routine collection and review of DB2 performance data
- Create and maintain historical performance data for trend analysis
  - Specific exception events
    - Event monitoring
    - Timeouts or deadlocks
    - EDM Pool full
    - Authorization failures
  - Thresholds exceeded
    - Elapsed, CPU, or wait times
    - Getpages and buffer updates

Is it Really a DB2 Problem?

Compare Total Task Elapsed Time Versus Accounting Class 1 Elapsed Time

Total Task Elapsed - Total time of a Batch Job or Total CICS/IMS response time

Class 1 Elapsed - First SQL call until thread termination

If A Large Difference Consider:

Application design and logic
Inefficient application initialization
Enqueue prior to DB2 thread creation

Poor CICS or IMS scheduling performance

Is it Really a DB2 Problem?

Compare Accounting Class 1 Elapsed to Accounting Class 2 In-DB2 Time

Class 1 Elapsed - First SQL call until thread termination

Class 2 In-DB2 - Time a task spent INSIDE DB2

If A Large Difference Consider:

- **Application design and logic**
- **Inefficient application code (outside DB2)**
- **Enqueue of resources outside of DB2**
- **Poor CICS or IMS performance**
- **Non-DB2 processing of rows retrieved**
- **Application & end user think time**
- **Thread wait for reuse time**

Look Outside DB2

What if Class 2 Time is Relatively Large?

High Accounting Class 2 In-DB2 Time

Class 2 In-DB2 - Time a task spent INSIDE DB2

If Relatively High Consider:

- Number and type of SQL calls being performed
- Poor SQL coding technique
- Poor index usage
- High DB2 Getpage counts
- Large DB2 scans and high I/O counts
- DB2 Lock/latch delays
- Large DB2 sorts
- Too many columns returned by SQL

Look Inside DB2

What is the Breakdown of Class 2 Time?

Compare Class 2 In-DB2 Time Relative To Class 2 In-DB2 CPU Time

Class 2 In-DB2 - Time a task spent INSIDE DB2

Class 2 In-DB2 CPU Time - CPU Time a task spent INSIDE DB2 doing SQL etc.

If CPU A High Percentage Consider:

- Poor SQL coding technique
- Number and type of SQL calls being performed
- Large amount of dynamic SQL
- Number of columns returned by SQL
- Unnecessary rows retrieved by SQL
- EDITPROC and FIELDPROC processing
- Large Sorts
- Too frequent commits
- Excessive locking

Look Inside DB2

Application CPU Tuning

- Minimize the number of SQL calls
  - SELECT is more efficient than OPEN, FETCH, CLOSE if only retrieving one row
- Filter unnecessary rows by adding predicates
  - More efficient than using program logic
- Use DB2 column functions versus program logic
- Minimize number of columns retrieved, updated, or inserted
- Minimize number of rows searched
- Use Static SQL versus Dynamic SQL
  - Example - SQLJ (static) versus JDBC (dynamic)

Is DB2 Waiting for Something?

Compare Class 2 In-DB2 Time Relative To Class 2 In-DB2 CPU Time

Class 2 In-DB2 - Time a task spent INSIDE DB2

Class 2 In-DB2 CPU Time - CPU Time a task spent INSIDE DB2 doing SQL etc.

If A Large Difference Consider:

- **Various possible delays**
- **DB2 measured (Class 3) delays**
  - **Poor index usage**
  - **High DB2 Getpage counts**
  - **Large DB2 scans and high I/O counts**
  - **DB2 Lock/latch delays**
  - **Large DB2 sorts**
- **Unaccounted for delays**

Look Inside DB2

DB2 Wait Counters

Compare Class 2 In-DB2 Time Versus Class 3 Wait Times

Class 2 In-DB2 - Time a task spent INSIDE DB2

Class 3 Wait Times - Time DB2 spent waiting for various events

Review Class 3 Wait Counters:

- **I/O Wait Time counters**
- **Lock/Latch Wait Time counters**
- **Service Task Wait Time counters**
- **and others…**

Look Inside DB2

High I/O Wait Counters

High Synchronous I/O Wait Time

High Asynchronous I/O Wait Time

High I/O Wait Time Consider:

- **Poor SQL coding technique**
- **Poor index usage**
- **High DB2 Getpage counts**
- **Large DB2 scans and high I/O counts**
- **Large DB2 sorts**
- **DASD and I/O contention**
- **Poor Virtual Pool performance and sizing**

I/O Wait Time Tuning

- Application I/O tuning
  - Review SQL and perform SQL optimization
  - Reduce number of pages of data accessed
- Eliminate System related I/O problems
  - DASD service times and DASD contention
  - Buffer Pool tuning issues (size, number, hit ratio)
- Database design
  - Review Index design, number of Indexes and Index placement

High Lock / Latch Wait Counters

High Accounting Class 3 Lock/Latch Wait Time

High Lock/Latch Wait Time Consider:

- **Needlessly large number of locks taken**
- **Locks being held for a long duration**
- **Lock escalation**
- **Commits too infrequently**
- **Lock timeouts and deadlocks**
- **Mix of concurrent applications**
- **Sequence in which applications update**

Lock / Latch Wait Time Tuning

- Lock / Latch Contention implies concurrency and the mix of applications being executed
- Review application BIND options
- Is Lock Avoidance being exploited?
- Can UR be employed?
  - Remove locking overhead

Is There Unaccounted Time?

Compare Class 2 In-DB2 Time Versus

The Sum of Class 3 Wait Time and Class 2 CPU Time

Class 2 In-DB2 - Time a task spent INSIDE DB2

Class 3 Wait Times - Time DB2 spent waiting for various events

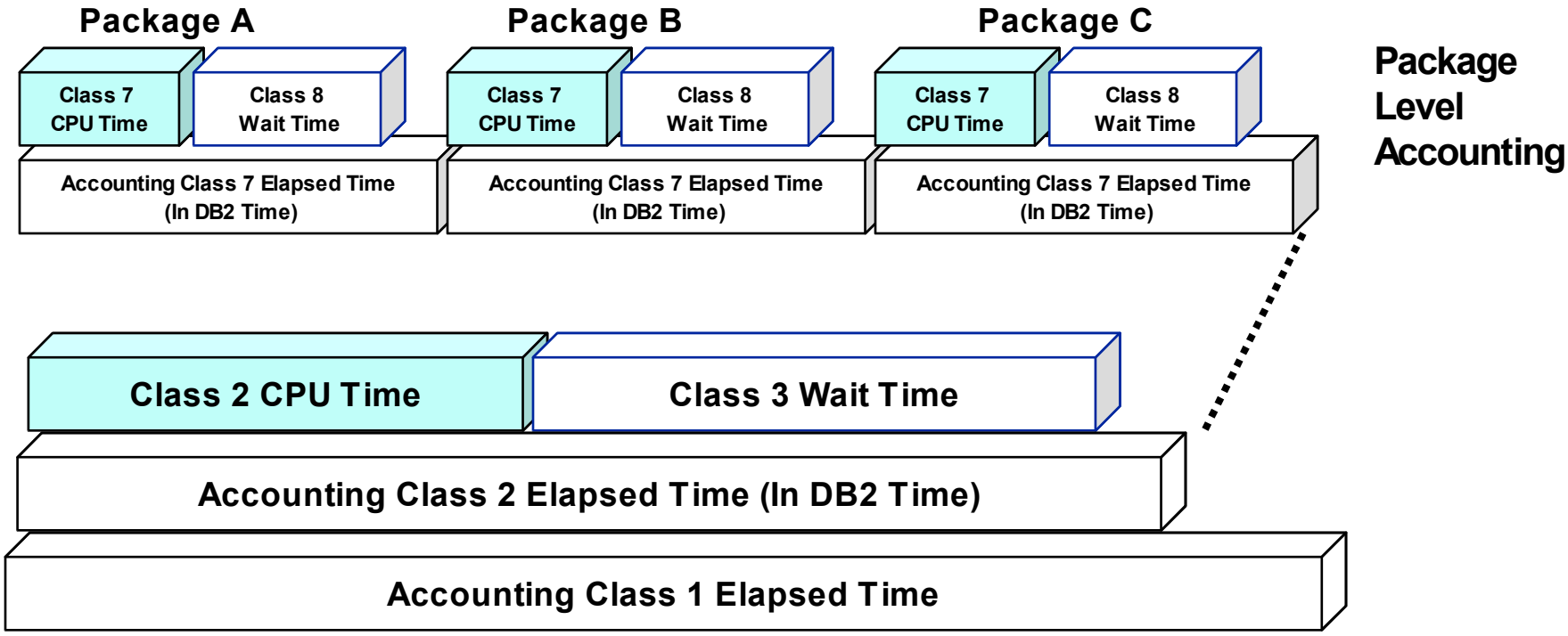
Class 2 In-DB2 CPU Time - CPU Time a task spent INSIDE DB2 doing SQL etc.

If A Large Difference Consider:

- **MVS dispatching priority**
- **MVS paging delays**
- **Operating system delays impacting DB2…**

Look Outside DB2

Is More Detail Needed?



- Accounting Class 7 & 8 provide package level detail
- SPs, UDFs, and triggers are all packages

Is Even More Detail Needed?

- Use Performance Traces to isolate further
  - Statement level
  - Detail activity within the statement level
- The most detailed level of tracing activity
- Use judiciously and with caution
  - Understand what is being traced
  - Trace only what is needed to isolate the problem
- Consider application flow
  - Is it a statement that is executed once but gets a million rows?
  - Is it a statement that is gets only one row, but is executed a million times?
- Find the offending SQL statement and perform DB2 Explain analysis

Tuning Summary

- Leverage your efforts - make the easy to implement changes first
  - Tune the z/OS environment
    - Workload management (WLM) policy
    - I/O subsystem
  - Change the data design
    - Create views
    - Add / remove / modify indexes
  - Modify application code
    - Tune SQL calls
  - Implement data purge / archive

- DB2 v8 Administration Guide SC18-7413
  - Part 5 Performance Monitoring and Tuning

Real-Time Monitoring with the OMPE 3270 Classic Interface

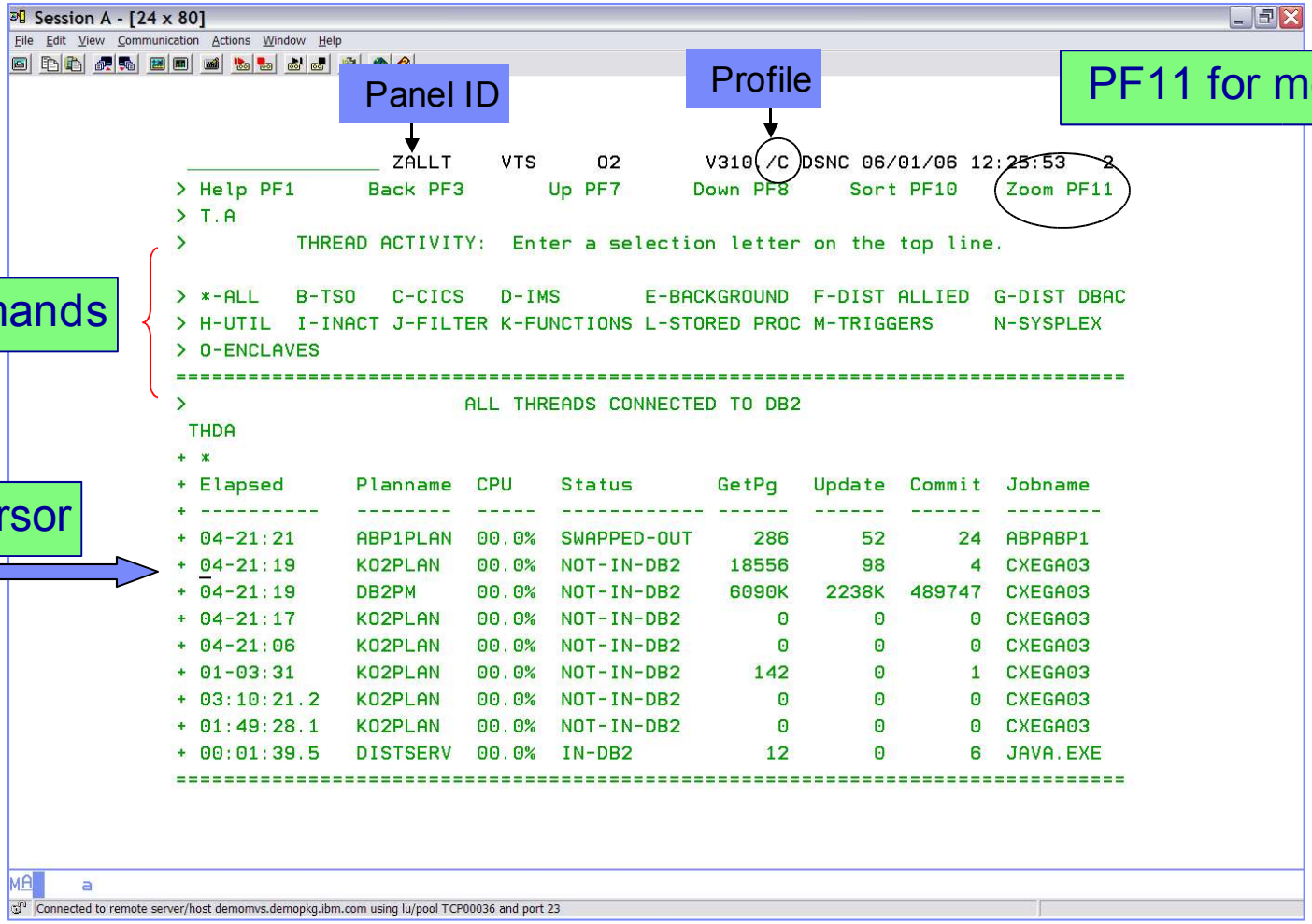
IBM Software Group

ON DEMAND BUSINESS™

Advantages

- Fast and Flexible
  - Easy to use the interface once trained on navigation
  - Get almost instantaneous response times due to the direct VTAM connection
- Reliable
  - Only dependent on a direct connection to OMPE server
- Full range of online capabilities
  - Real time monitoring
  - Object analysis
  - Near term history
  - Application Trace Facility
  - DB2 Connect
  - Monitoring of locking contention
  - IFCID tracking

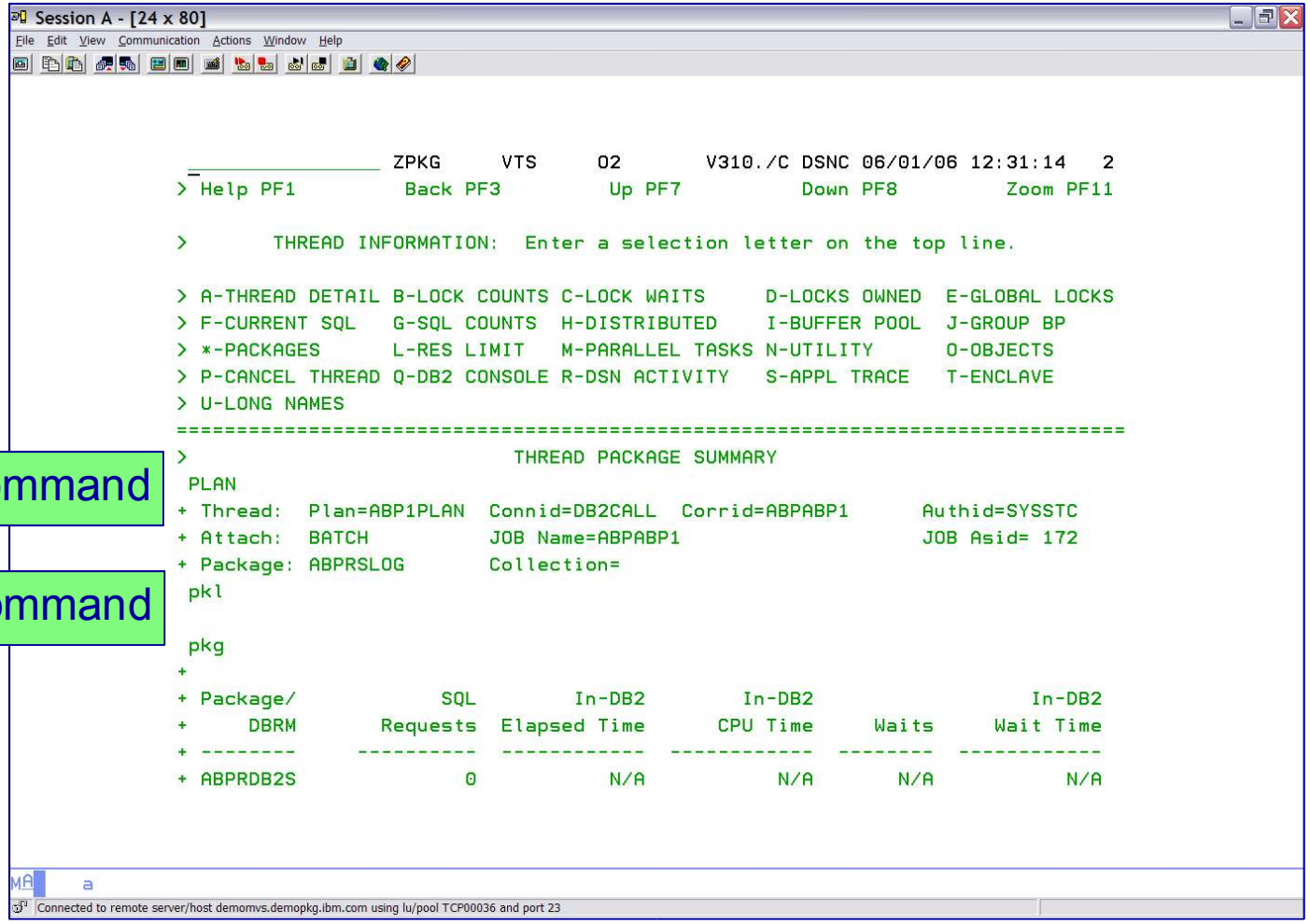
Sample Panel



Major and Minor Commands

Major Command

Minor Command



Major and Minor Commands

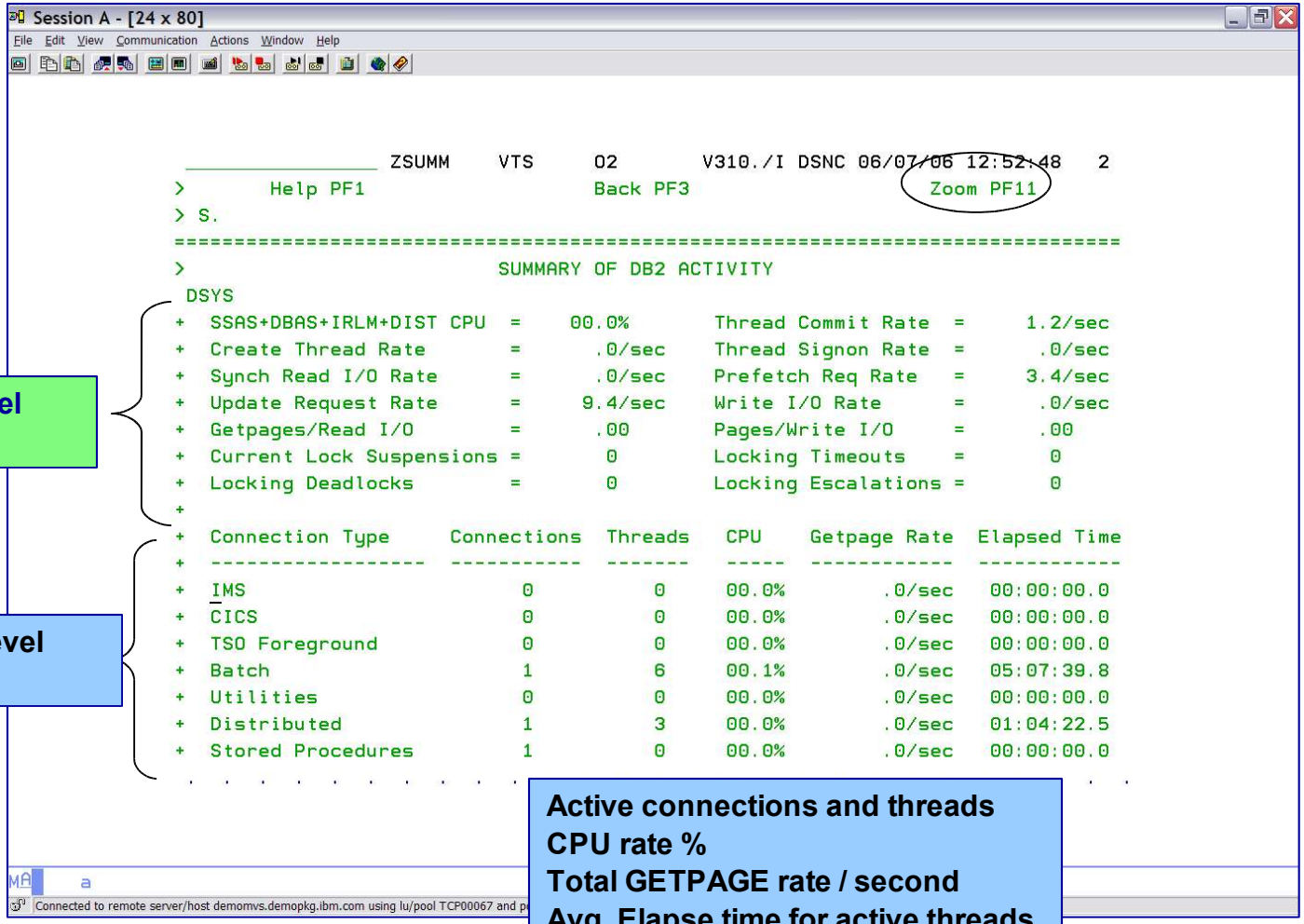
- Major Commands
  - Displayed in uppercase
  - A major command preceded by a blank can be used on any panel to return information related to the command
  - Some major commands must be followed by a parameter
- Minor Commands
  - Displayed in lowercase
  - Cannot be used alone
  - Can only be used on panels with the major commands

Summary of DB2 Activity - Statistics Data

- Critical system wide activity and resource utilization
  - Main Menu option S
- Collected by collection type – IMS, CICS, TSO, Batch, Utilities, Distributed, and Stored Procedures
- Summary of activities grouped by subsystems connected to DB2
- Rates are calculated from the last OMPE display
- All values are based on active threads
- Highlighted fields represent exceptions (red)
- ZOOM (PF11) is active for this panel
  - Place cursor on the connection type and press PF11

Each time you press ENTER your screen is refreshed

Summary of DB2 Activity - Statistics Data



System Level information

1 line system level summary

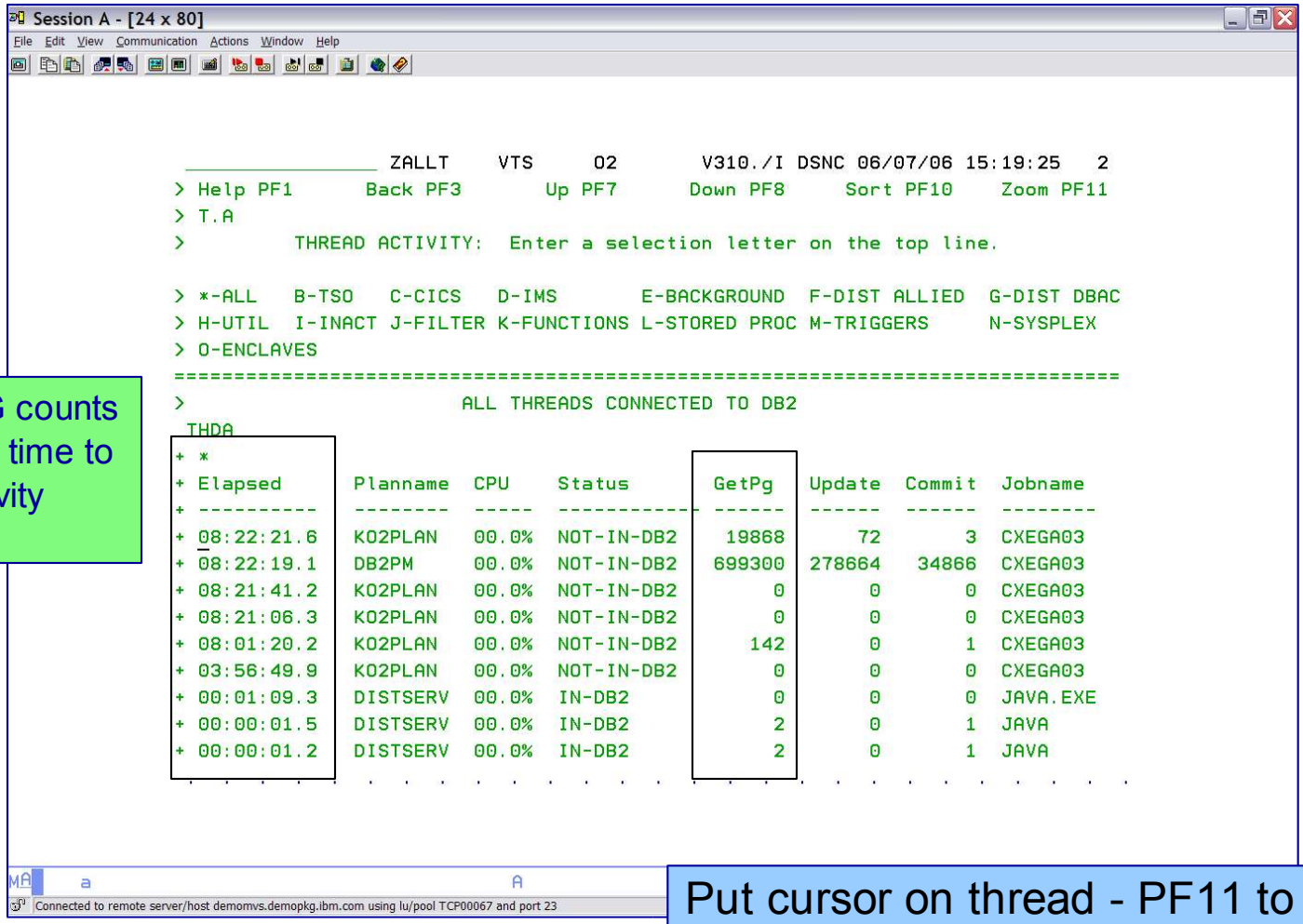
Active connections and threads
CPU rate %
Total GETPAGE rate / second
Avg. Elapse time for active threads

Thread Activity - Accounting Data

- Information on ALL connected threads
  - Main Menu option T - by plan (Accounting classes 1,2,3)
  - Main Menu option U - by package (Accounting classes 7,8)
- Each row displays information about an individual thread
  - Filtering available
- Columns are ordered by thread elapsed time
- Includes information about
  - DB2 and MVS resource consumption
  - DB2 activity
- Place cursor on any thread - PF11 (ZOOM) to display Thread Detail

Thread Summary Activity - Accounting Data

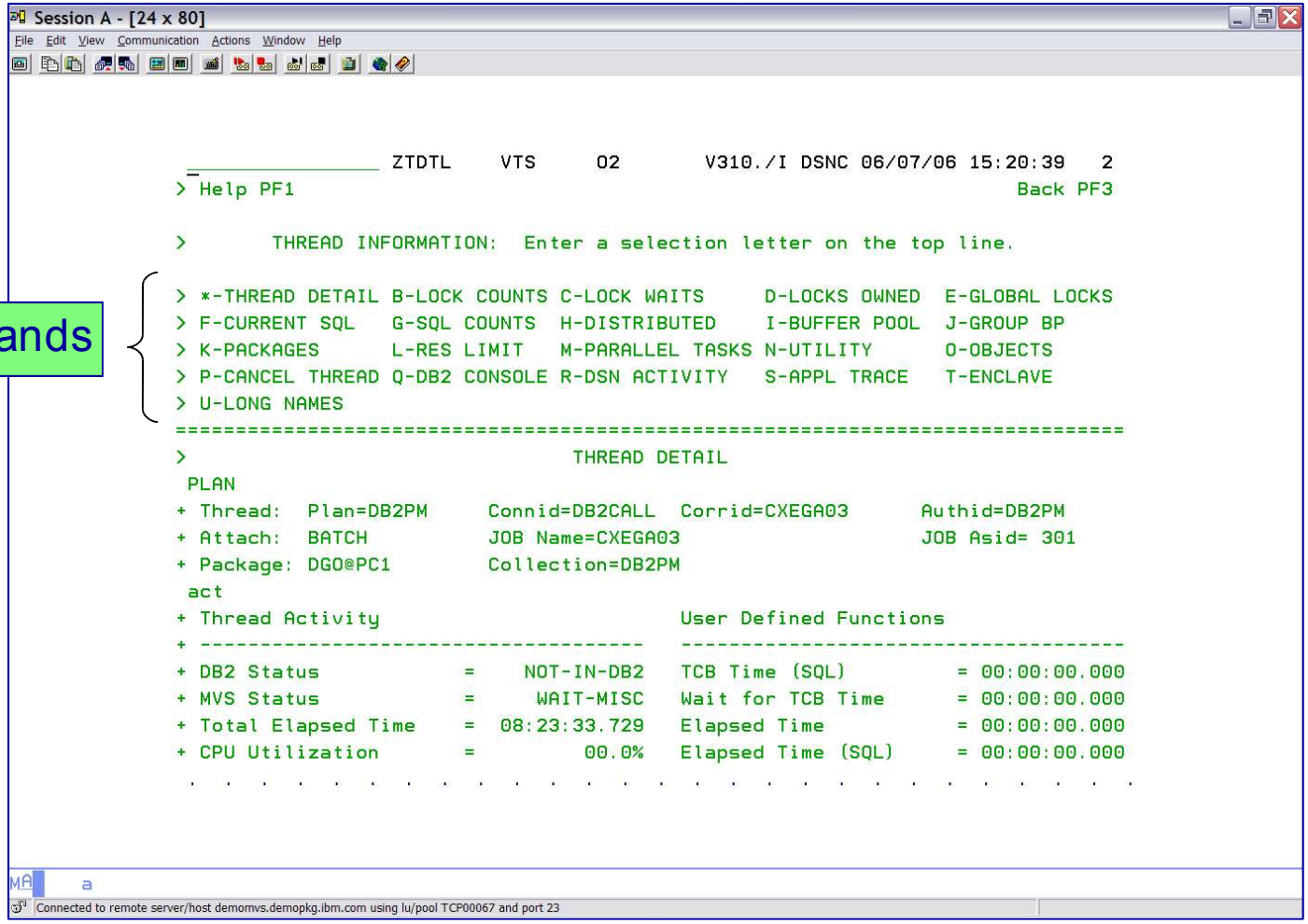
Look at the GETPG counts and thread elapsed time to determine high activity threads



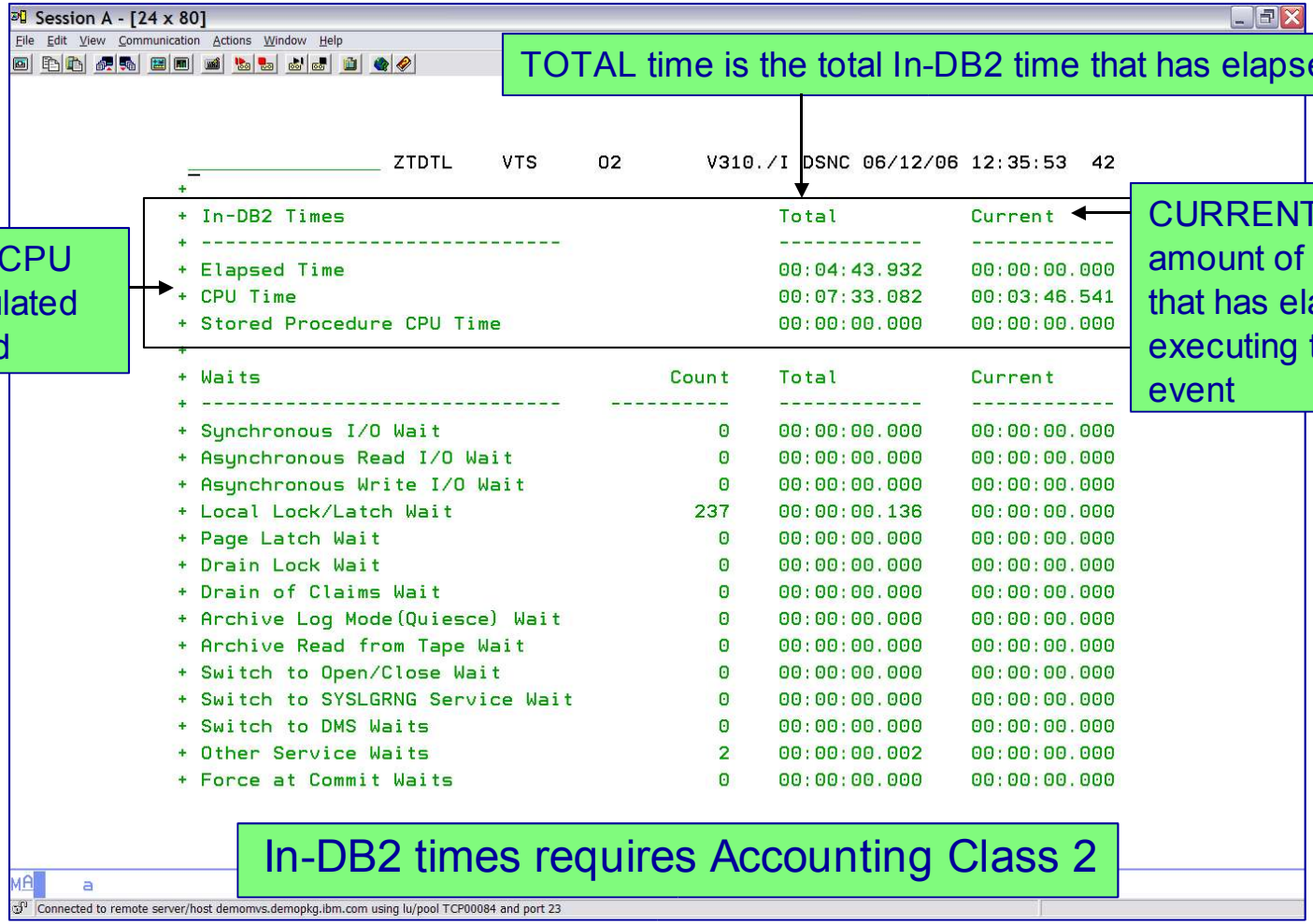
Put cursor on thread - PF11 to get to Thread Detail

Thread Detail Activity - Accounting Data

Commands



Thread Detail Activity - Accounting Data



TOTAL time is the total In-DB2 time that has elapsed

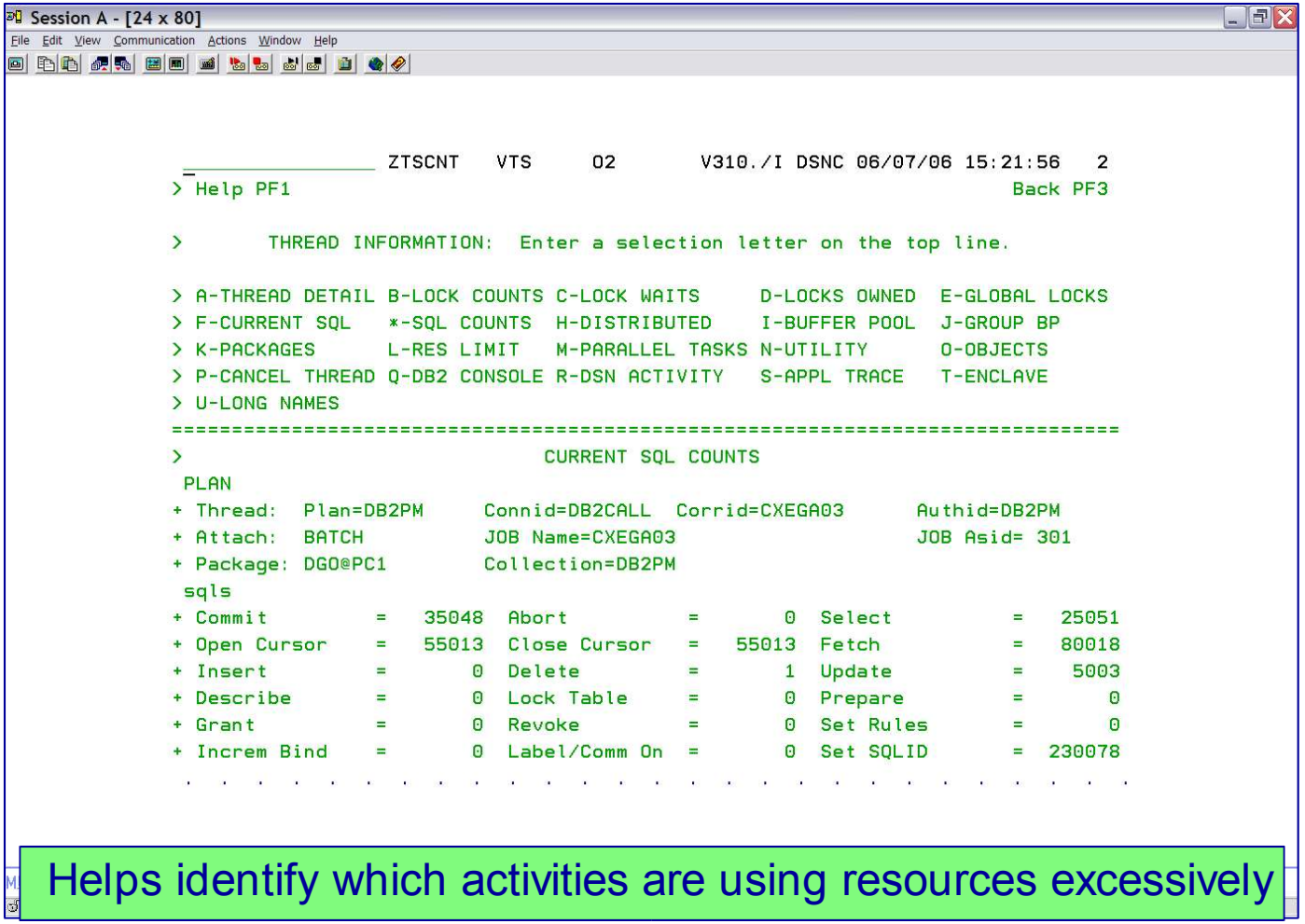
CPU is total CPU time accumulated for the thread

CURRENT is the amount of In-DB2 time that has elapsed executing the current event

In-DB2 times requires Accounting Class 2

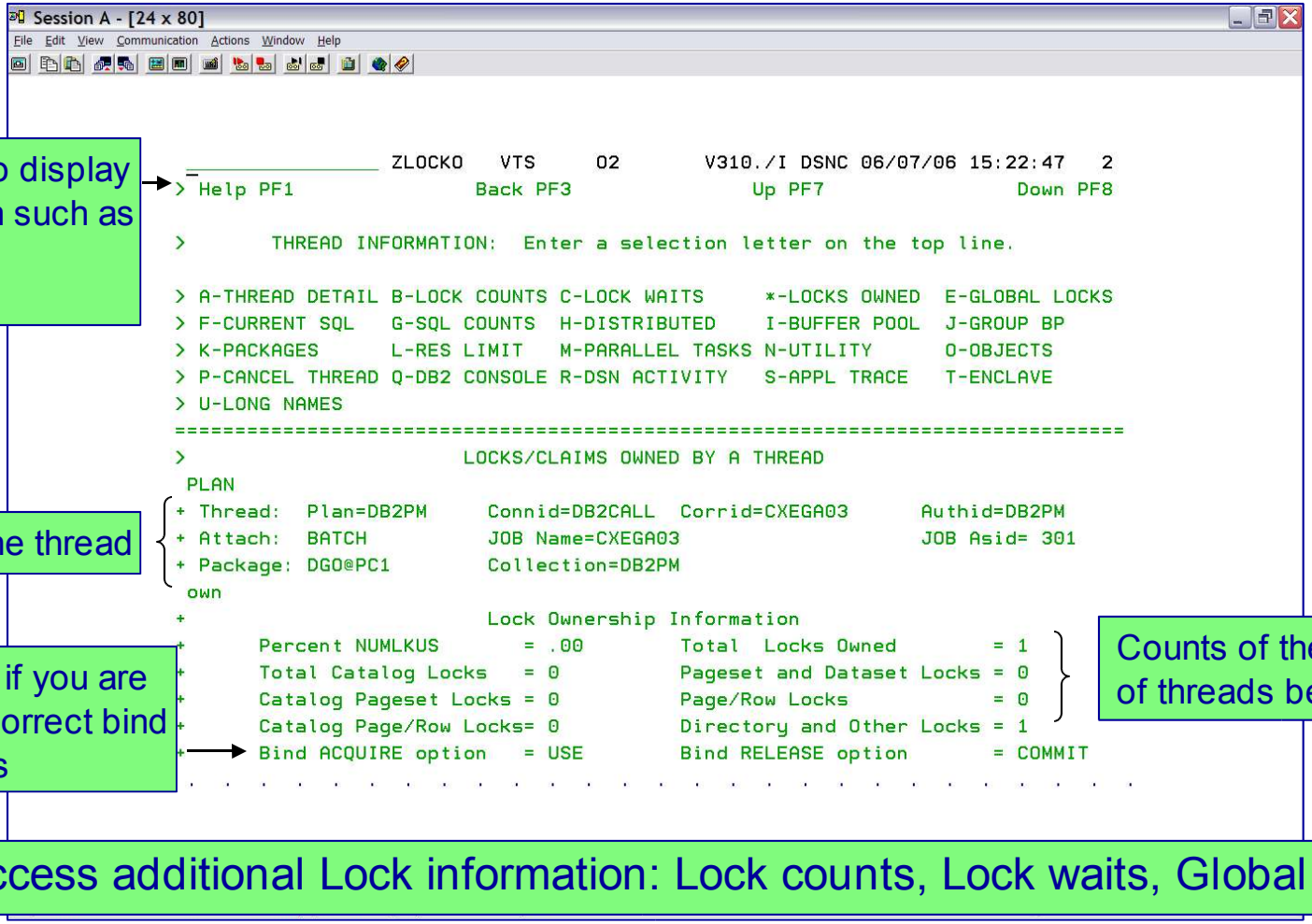
Wait times requires Accounting Class 3

Thread Detail Activity - SQL Counts (G)



Helps identify which activities are using resources excessively

Thread Detail Activity - Locks Owned (D)



Use PF1 to display information such as lock types

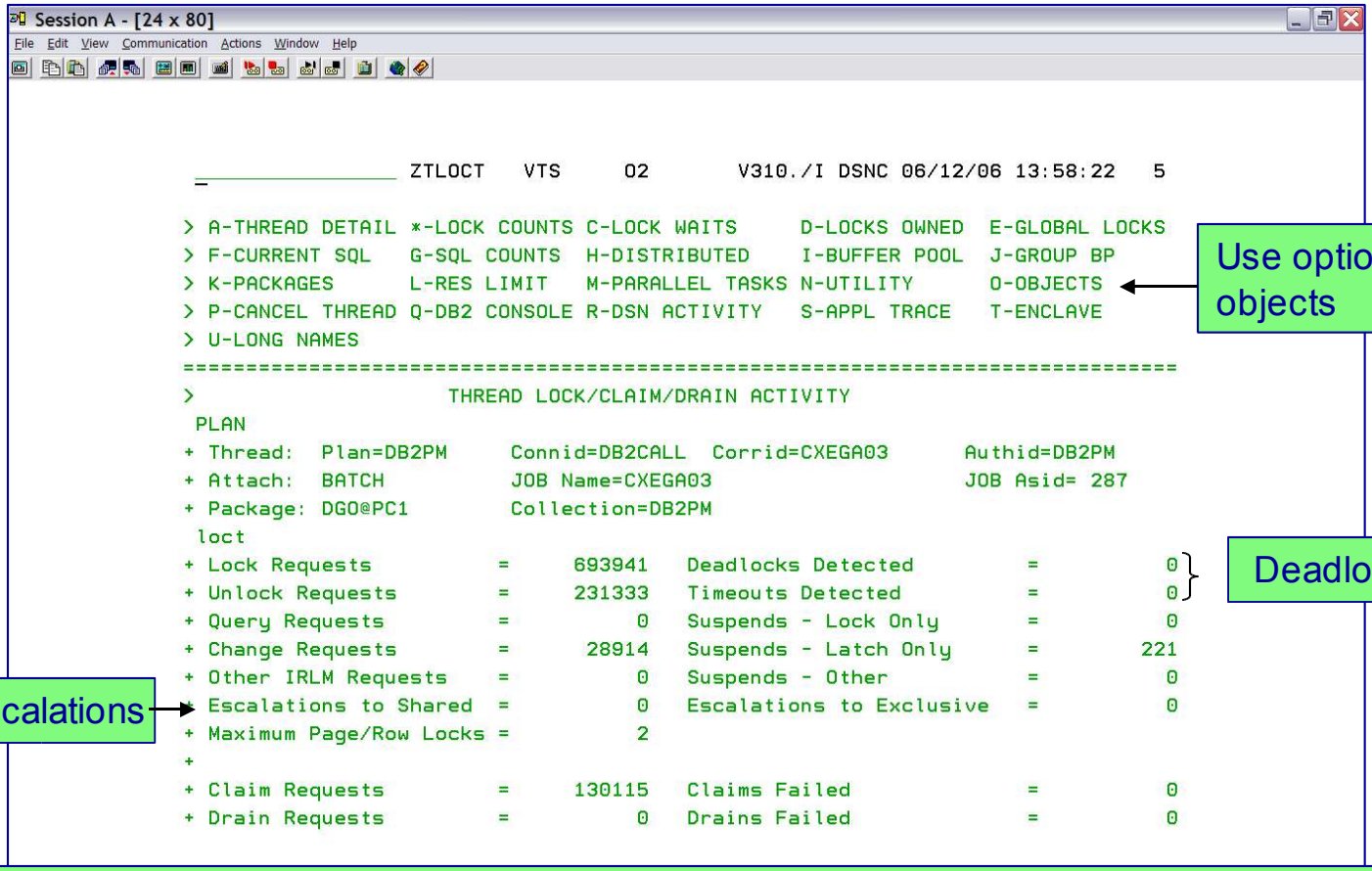
Identifies the thread

Determine if you are using the correct bind parameters

Counts of the types of threads being held

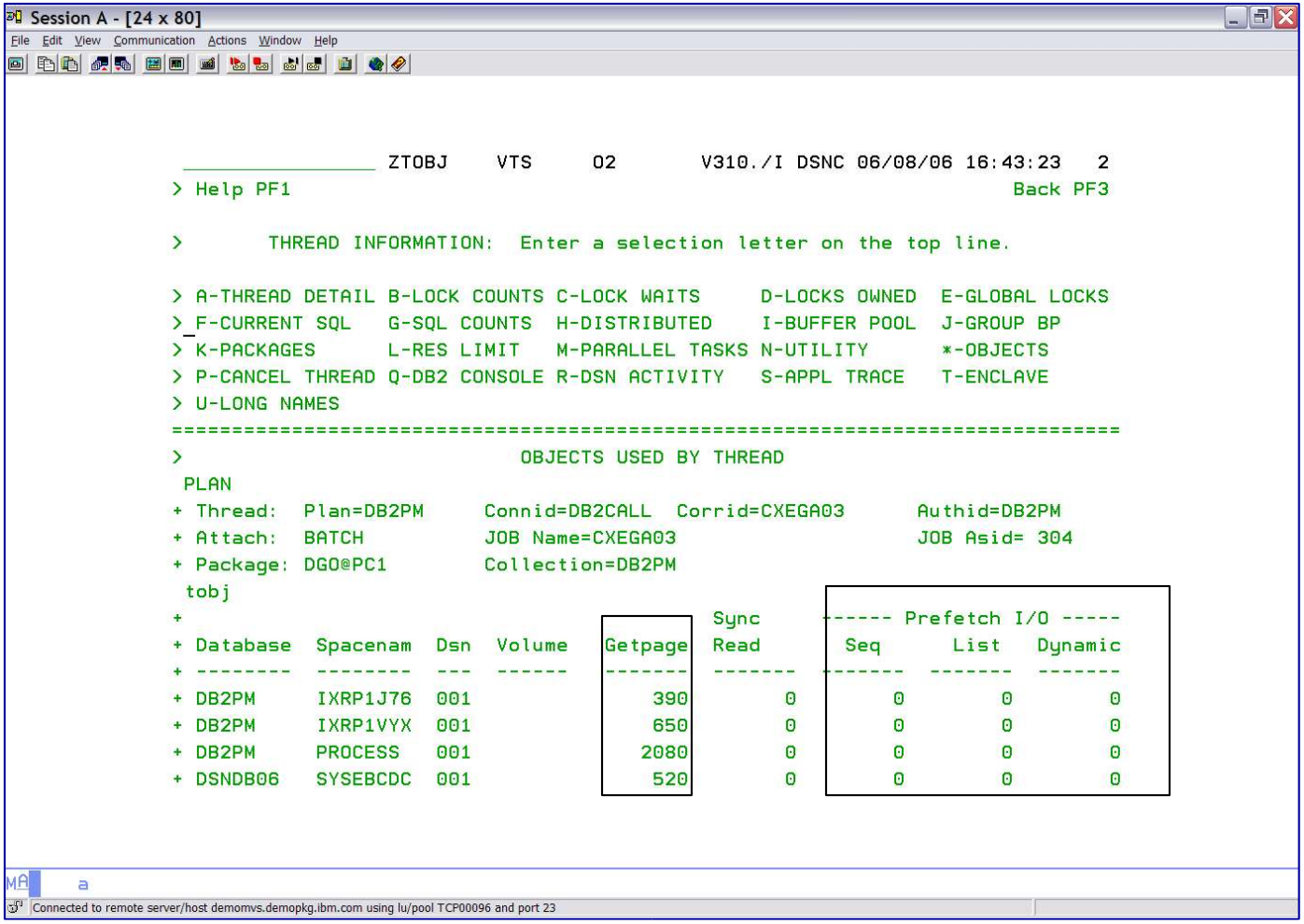
Can access additional Lock information: Lock counts, Lock waits, Global Locks

Thread Detail Activity - Lock Counts (B)

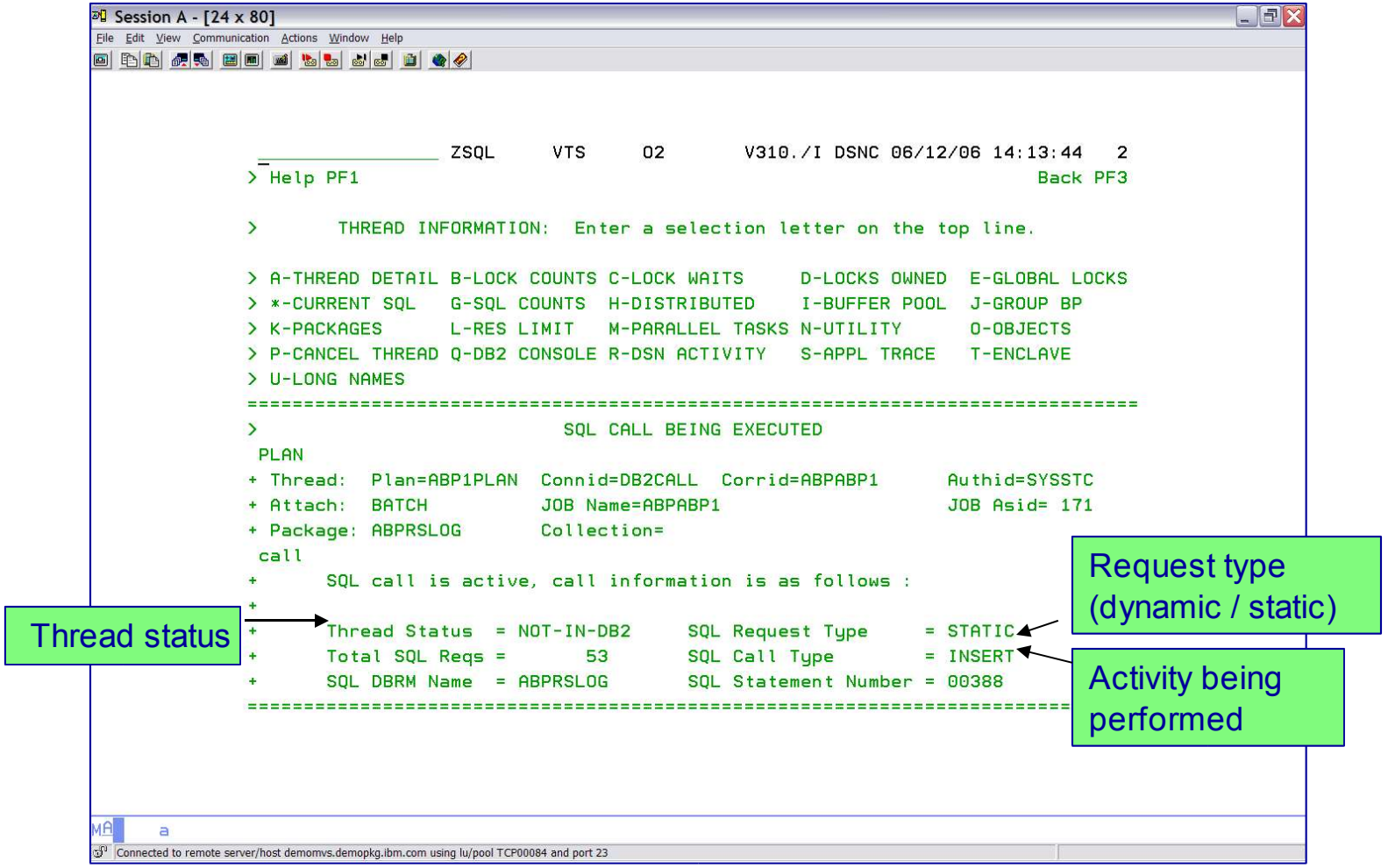


Drain - action to acquire a locked resource by quiescing access to the object
Claim - informs that the object is being accessed; no other object can be drained until a commit is reached

Thread Detail Activity - Objects (O)



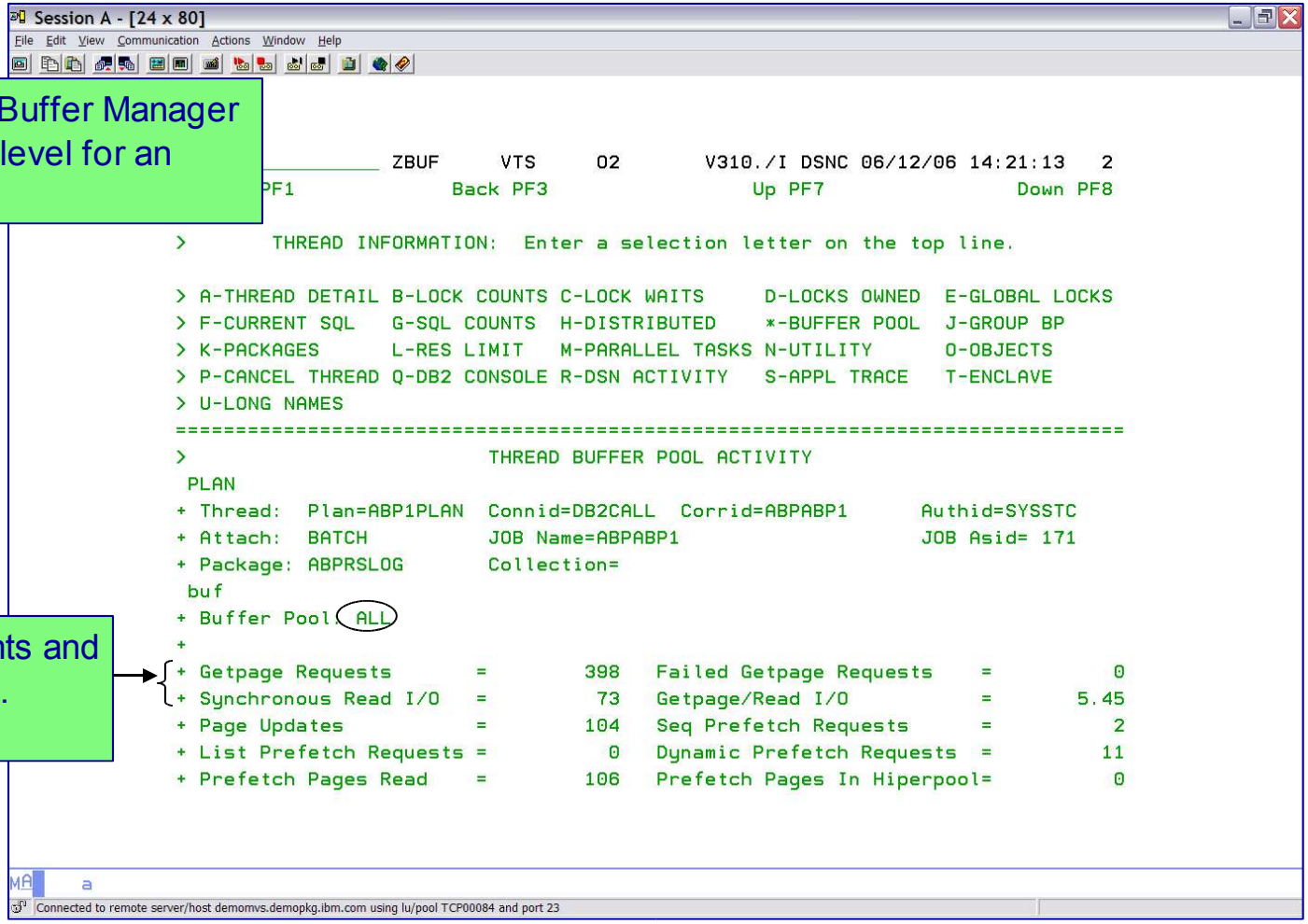
Thread Detail Activity - Current SQL (F)



Thread Detail Activity - Buffer Pool (I)

Information about Buffer Manager activity at the pool level for an individual thread

Note Getpage counts and synchronous I/O vs. Prefetch requests

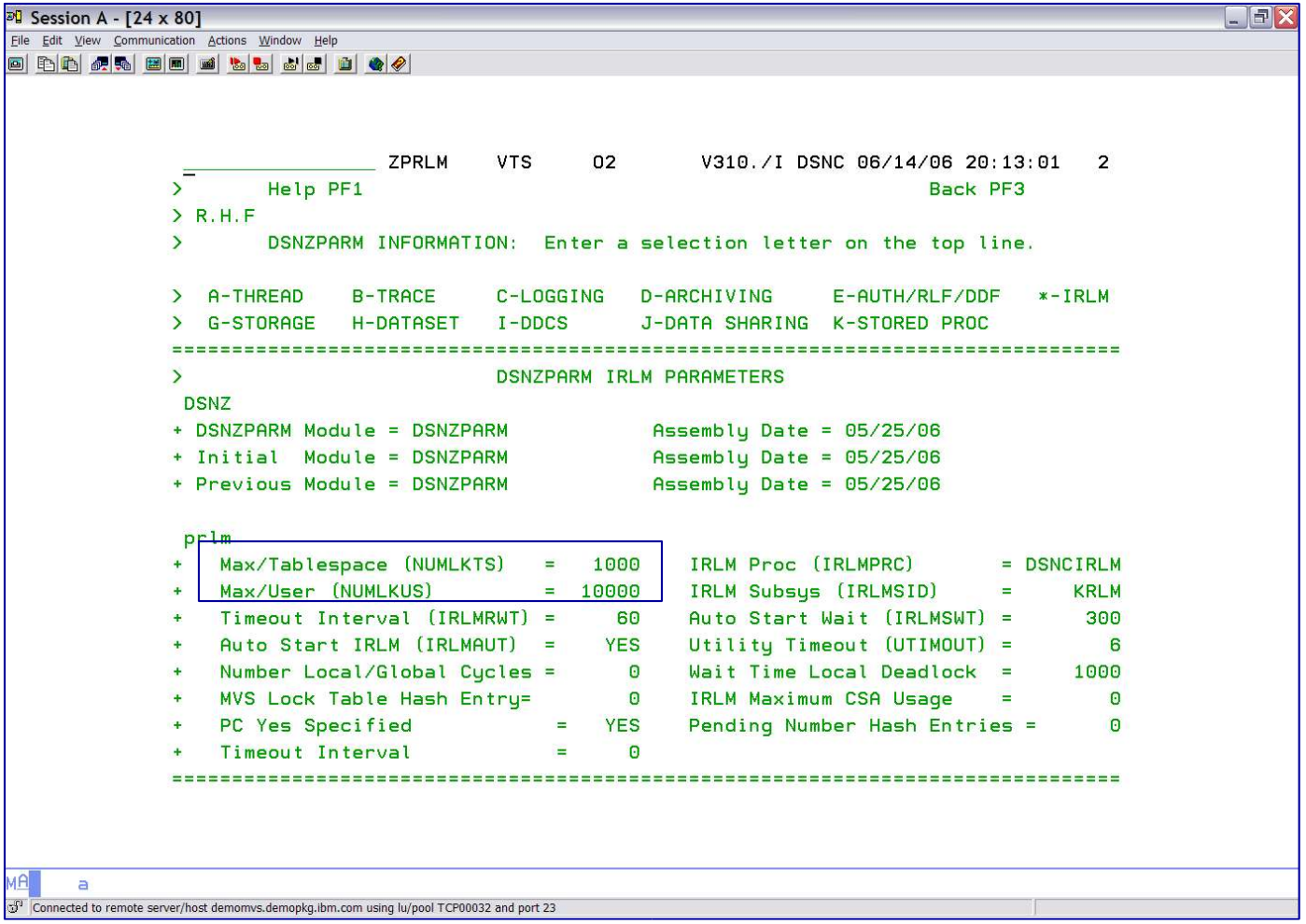


Lock Escalation

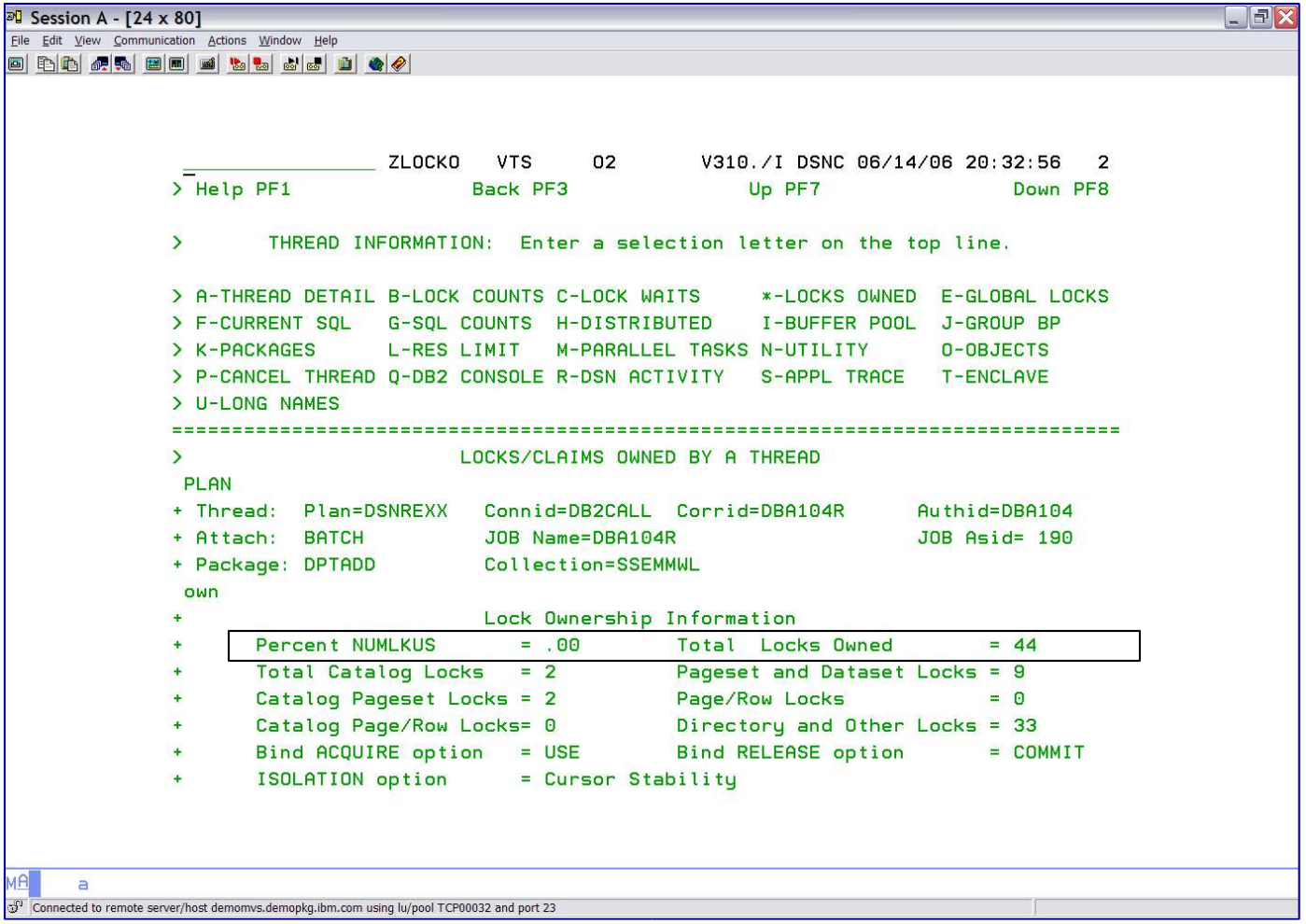
- Occurs typically in batch jobs that have heavy update activity and commit infrequency
  - DSNZPARM parameters
    - NUMLKTS - default value for max # page / row locks a thread can hold on a single table space
      - Used when TS is defined with LOCKMAX=SYSTEM
      - When limit is reached - escalates to single tablespace lock (unless tablespace is defined with LOCKMAX=0)
    - NUMLKUS - total # page / row locks across all tablespaces that a thread can hold (typical value is 20000)
      - -904 when limit is reached
      - Batch job abends
    - Lock escalation is preferable to -904

How can you see the values?
From the main menu R.H.F

DSNZPARM Settings



Lock Escalation



Lock Escalation

Session A - [24 x 80]

File Edit View Communication Actions Window Help

```
                  ZLOCKO    VTS      02       V310./I DSNC 06/14/06 20:35:43   25
_
+
+       Type     Level        Resource                                  Number
+       ----     -----        ------------------------------------ ------
+       DPAG         X        DB=SSEMMWL         PS=GLWS002                 1
+       DTBS         S        DB=DSNRGFDB                                   1
+                    S        DB=DSNDB07                                    1
+       PSET        IS        DB=DSNDB06         PS=SYSSEQ                  1
+                   IX        DB=TEMPDB          PS=TEMPTS                  1
+                    S        DB=DSNRGFDB        PS=DSNRGFTS                1
+                   IX        DB=SSEMMWL         PS=GLWS002                 1
+                    S        DB=DSNDB07         PS=DSN4K01                 1
+       SKCT         S        Plan=DSNREXX                                  1
+       PALK        IS        DB=SSEMMWL         PS=GLWSEMP                 4
+       MDEL         S        DB=SSEMMWL         PS=GLWSEMP                 1
+       TABL        IS        DB=DSNDB06         PS=SYSSEQ                  1
+                   IS        DB=DSNRGFDB        PS=DSNRGFTS                1
+                   IS        DB=DSNRGFDB        PS=DSNRGFTS                1
+                   IX        DB=SSEMMWL         PS=GLWS002                 1
+       SKPT         S        Coll=QEMNT6UXV        D Pkg=QSS7IEVE
+                             Token=179B1B7F010832EC                        1
+                    S        Coll=T!MEST!WS           Pkg=AESUAQT7
+                             Token=17BFB72605EECA33                        1
+                    S        Coll=T!MEST!WS           Pkg=ASTD!AS6
```

Identify which tablespaces are holding a large # page / row locks that have not been escalated
May have to reduce the LOCKMAX value below default value of NUMLKTS to prevent -904

Lock Escalation - Resolving -904

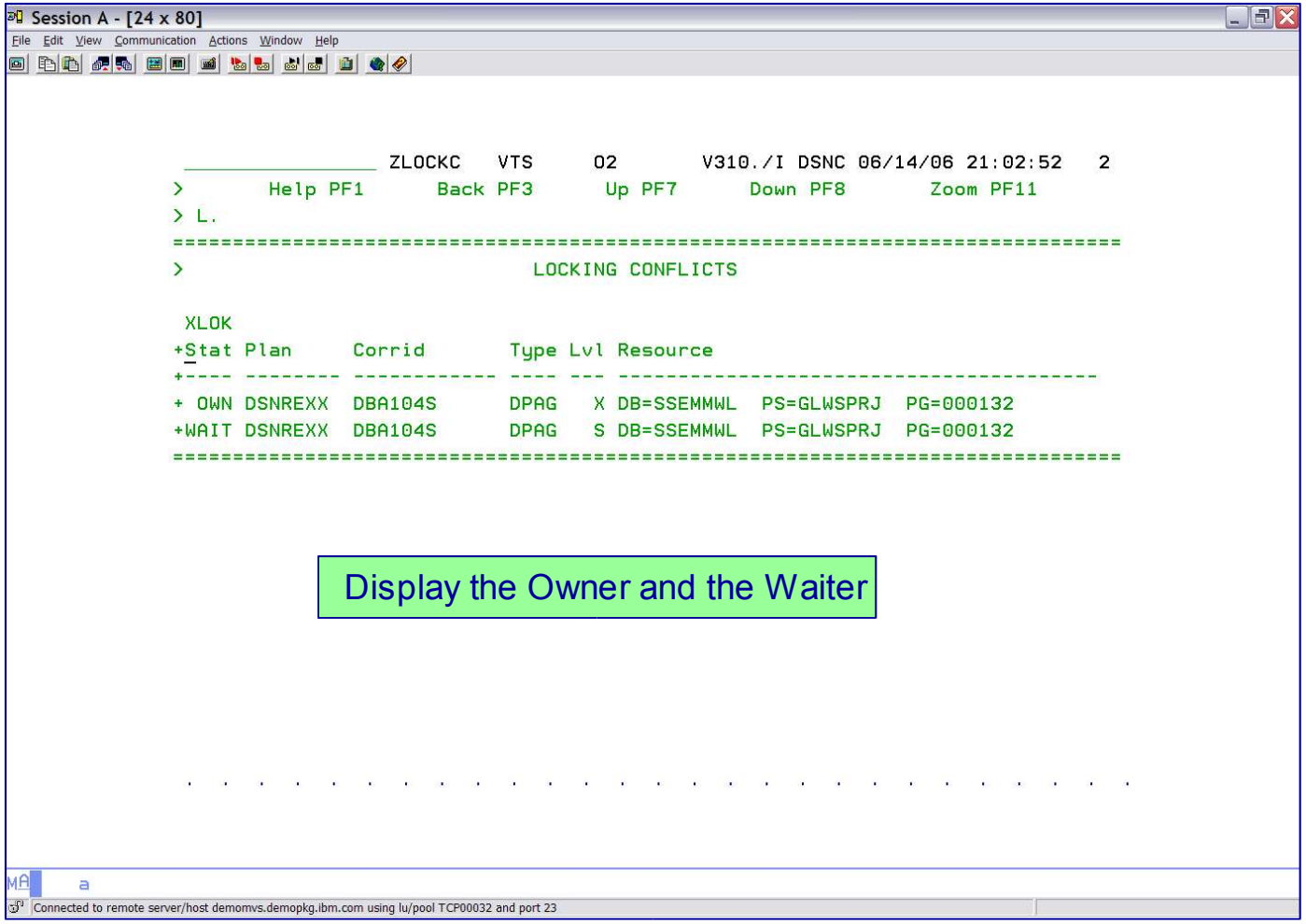
- Increase COMMIT frequency of the program
- Add COMMIT processing to the program if it has none
- Explicitly LOCK some of the tablespaces used by the program – LOCK TABLESPACE

Deadlocks / Timeouts

- Option L off of the main menu
- Position the cursor on a thread waiting for a lock - PF11 ZOOM to the Thread Detail panel
- On Thread Detail panel select option F to view SQL call waiting for the lock
- Determine why the lock wait is occurring

Alternative is to use Exception triggering with screen logging

Deadlocks / Timeouts



Application Traces

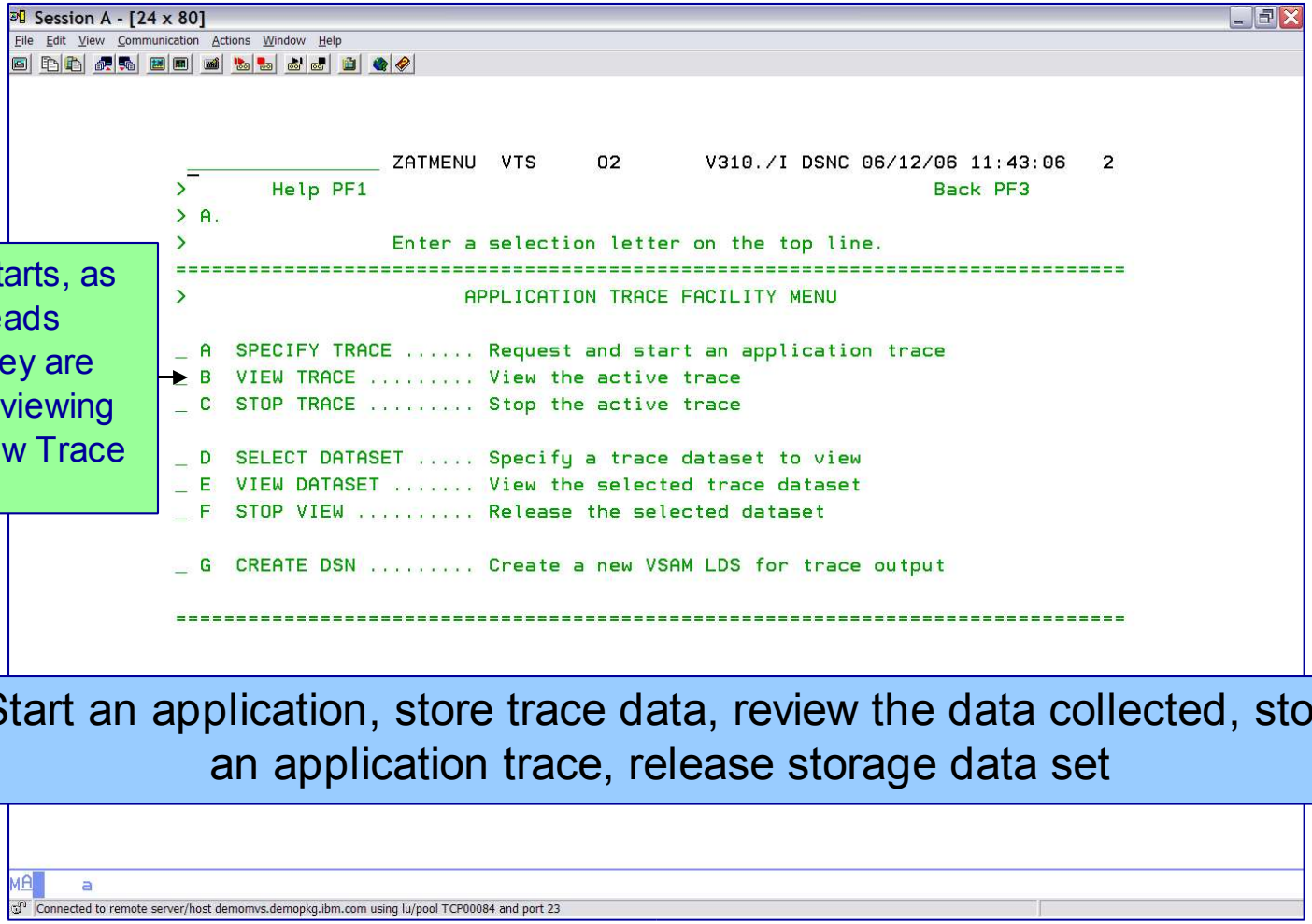
- Selective tracing of thread events associated with DB2 applications
- Provides a view into the performance portion of a workload and resource consumption
  - SQL trace information
  - Sort activity information
  - Pageset access and scan information
  - Locking information
  - Buffer activity
  - Application In-DB2 time and In-DB2 CPU time
- To monitor statement level and below

Application Traces

- Records diagnostic information in regards to events in VSAM file
- 6 types of traces: Accounting, Audit, Global, Monitor, Performance, Statistics
- Start traces
  - DSNZPARM
  - -START TRACE command
- Each trace is composed of IFCID's
  - Defines a record that represents a trace event
  - Smallest tracing unit in DB2
- Trace data can be stored in data spaces or pre-allocated VSAM linear data sets
  - ICAT creates a default set if ATF data sets
  - Suggestion is that each user create their own for concurrent tracing

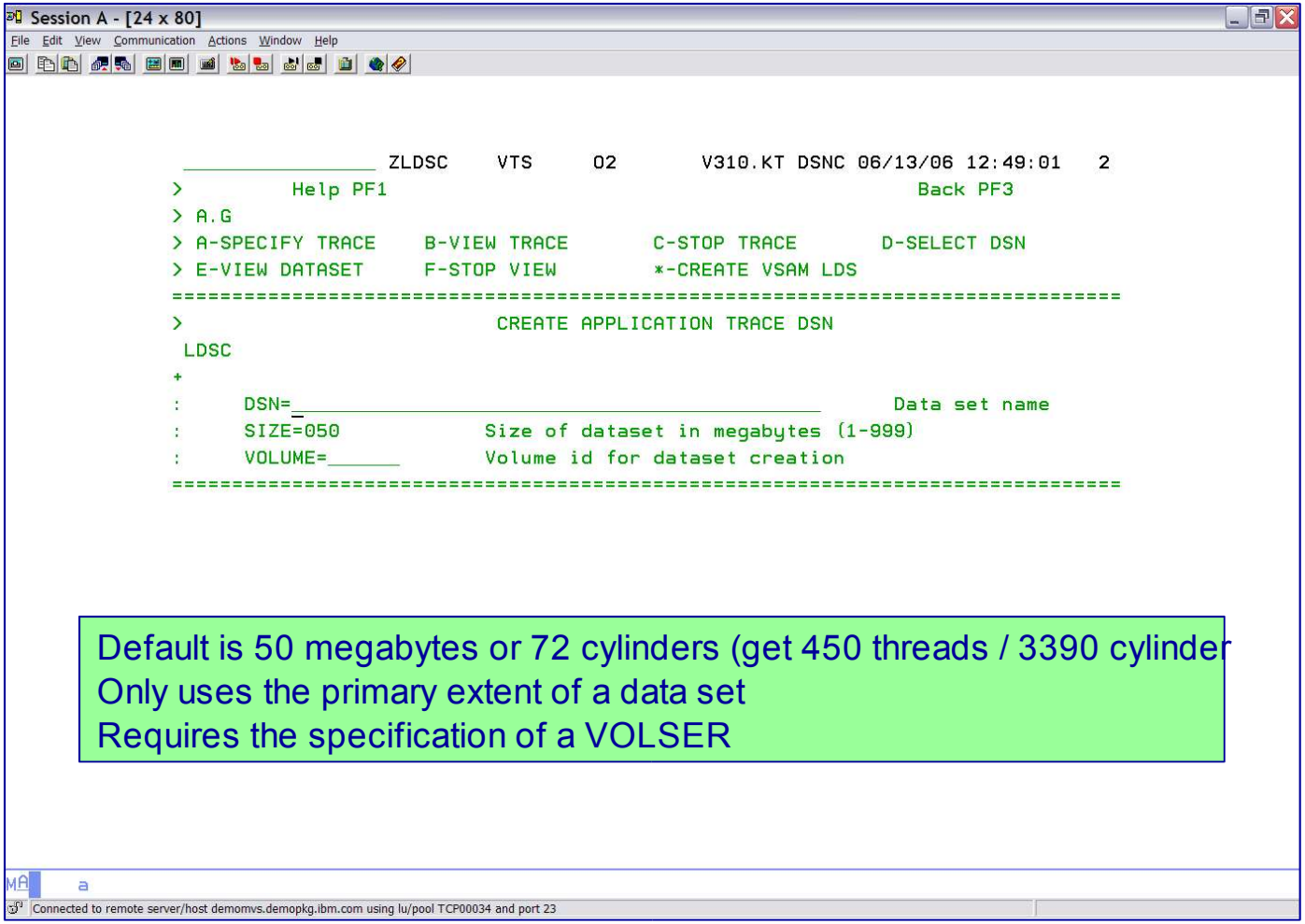
Application Trace Facility (ATF)

Once ATF starts, as qualified threads **complete** they are available for viewing from the 'View Trace Dialog'



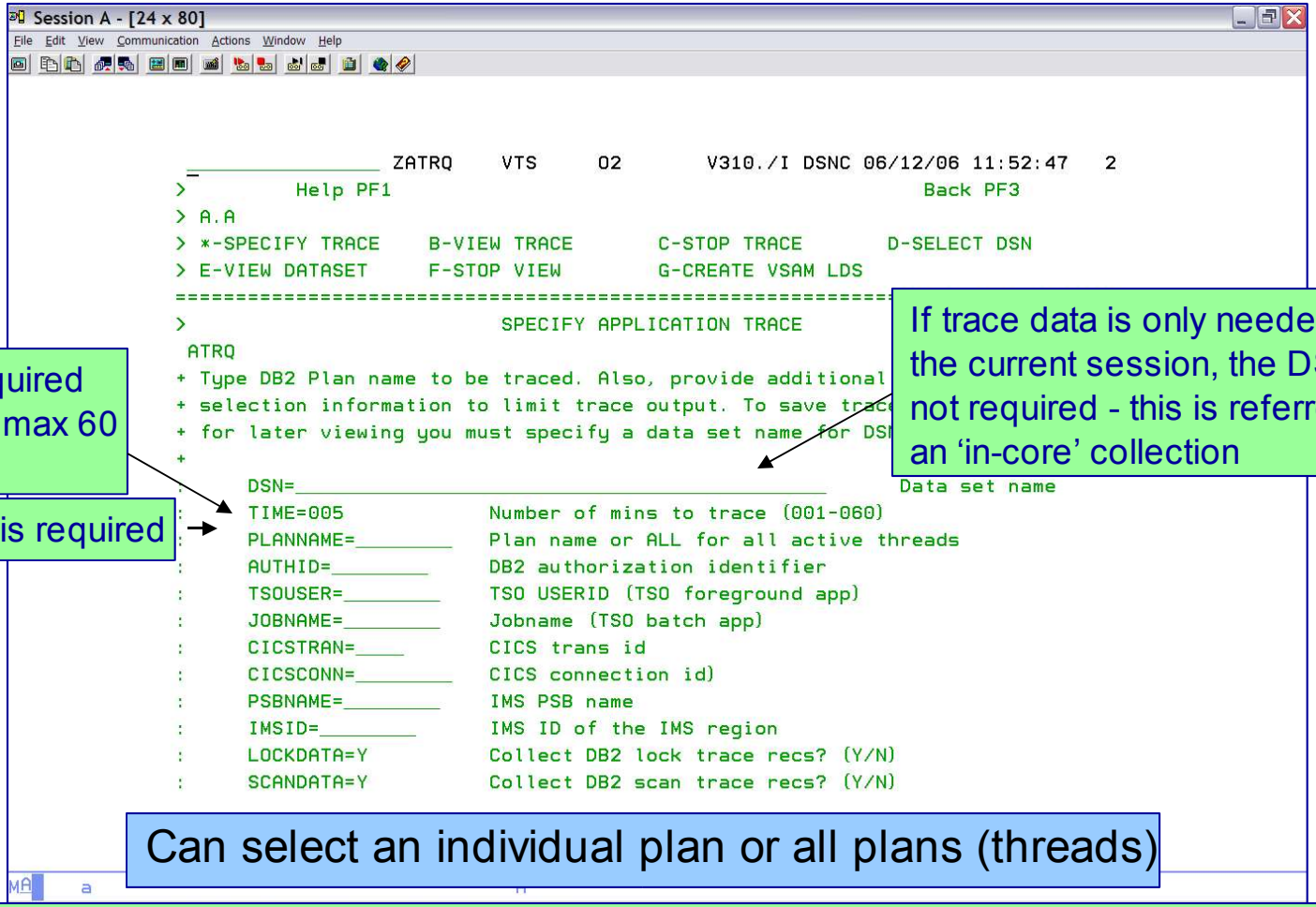
Start an application, store trace data, review the data collected, stop an application trace, release storage data set

ATF - Create DSN (G)



Default is 50 megabytes or 72 cylinders (get 450 threads / 3390 cylinder
Only uses the primary extent of a data set
Requires the specification of a VOLSER

ATF - Specify Trace Criteria (A)



Trace time is required
Default is 5 min; max 60 min

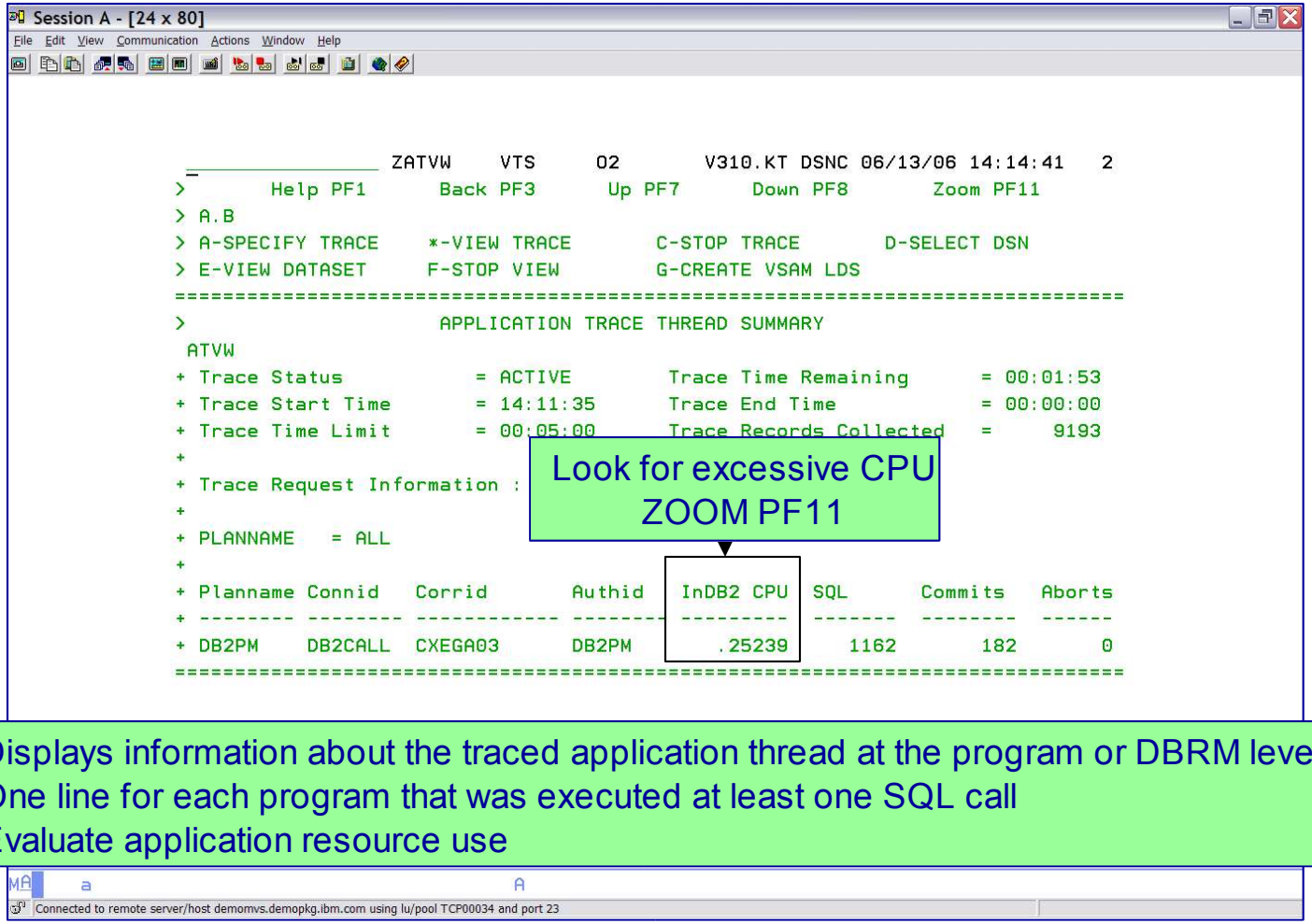
Plan name is required

If trace data is only needed for the current session, the DSN is not required - this is referred to as an 'in-core' collection

Can select an individual plan or all plans (threads)

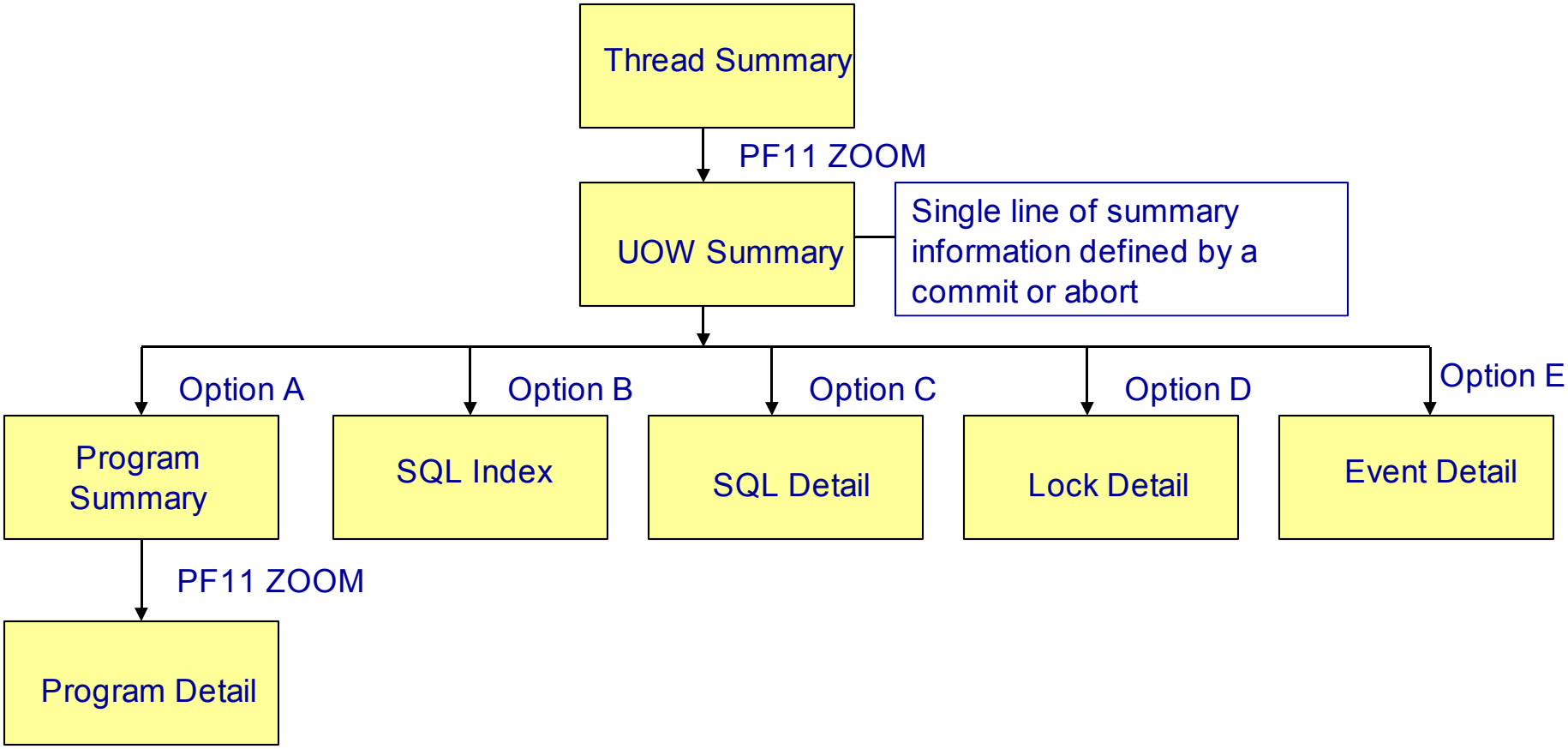
Recommendation from Redbook: Turn off LOCK DATA and SCAN DATA unless needed

ATF - View Trace (B)



Displays information about the traced application thread at the program or DBRM level
One line for each program that was executed at least one SQL call
Evaluate application resource use

ATF - Navigation



Near-Term History

- Storing of all statistics and accounting records written by DB2 over a recent period of time
- Accounting record is written by DB2 when the thread terminates
  - Unless the thread has terminated, there will be no entries in near-term history
- Could be collecting performance data at the thread level
  - Sorting / Dynamic SQL / Locking / Scanning
- Short-Term History vs. Near-Term History
  - Short-Term History is real time snapshot of the system is taken at regular intervals - e.g. 1 per minute - and stored
    - May miss short running threads
  - Near-Term History captures all threads that complete in the recording interval

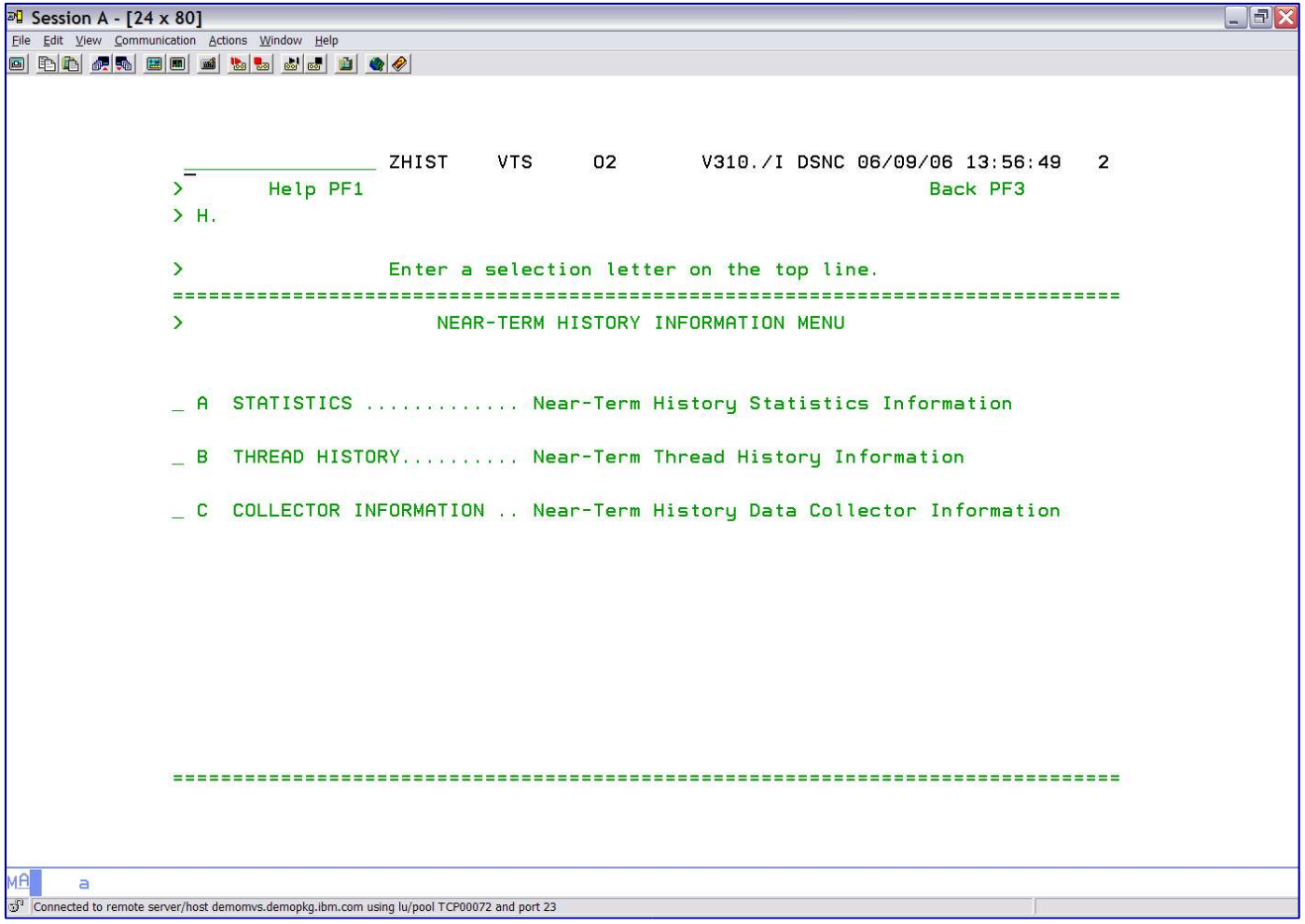
Near-Term History Benefits

- Some events are to short termed to be viewed in short term
- Deadlocks / timeouts may be resolved before you can look at them
- Batch jobs may be executing thousands of SQL calls / second
- Track performance problems for specific times
- Identification of threads that have experienced problems in the last few hours
  - Excessive CPU / Elapse time
  - Excessive in-DB2 CPU / Elapse time
  - Threads with timeouts / deadlocks
  - Threads committing to infrequently
  - Threads that have aborted
  - Threads with excessive lock waiting time
  - Threads with excessive DB2 wait time for I/O

Near-Term History Management

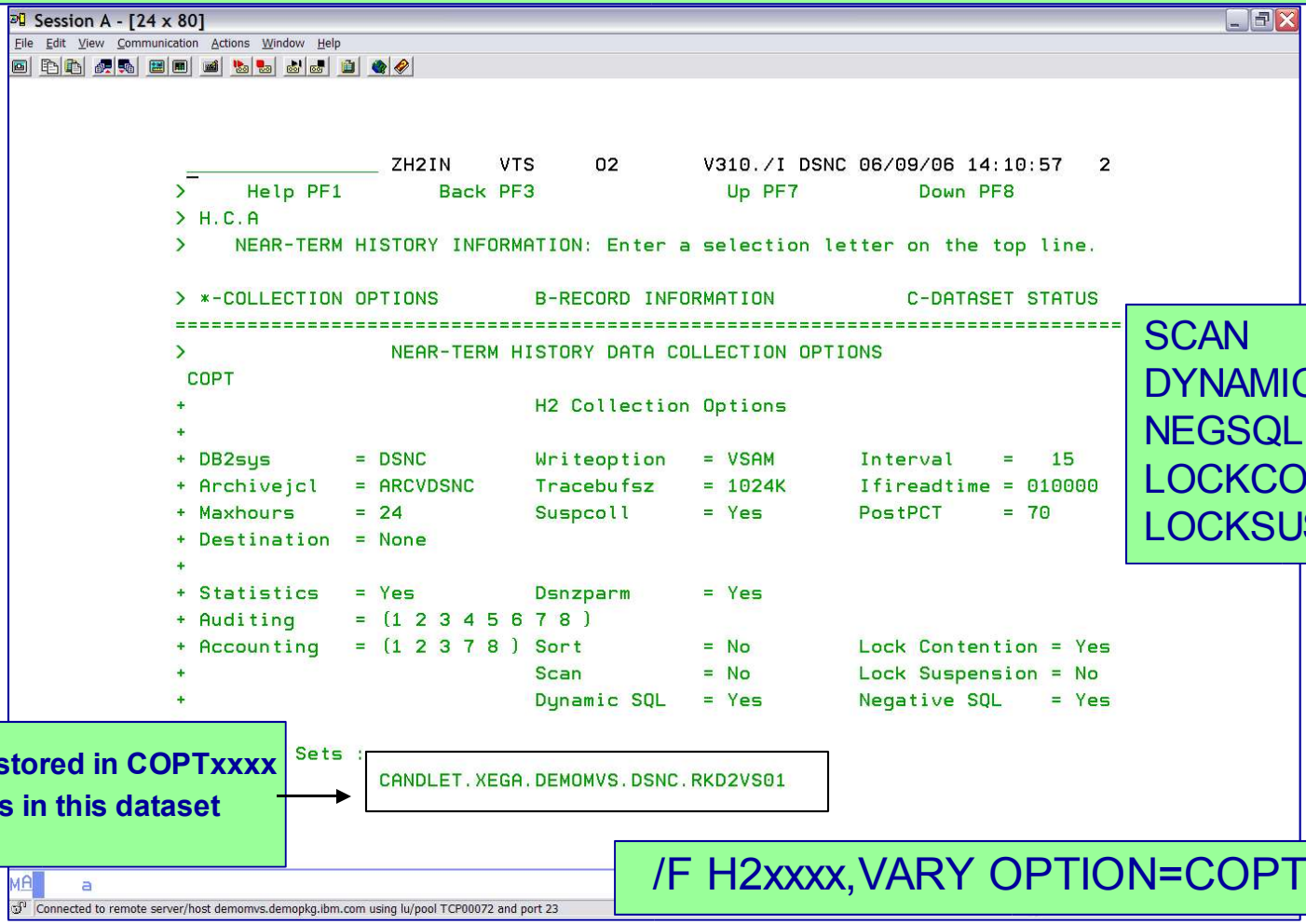
- Must archive in order to empty the VSAM data sets
- ICAT defaults to 3 near term VSAM datasets
  - Can increase to 10
- The amount of history depends on how active your system is and how large the history data sets are
- Only stores dynamic SQL calls, and only if the collector option for dynamic SQL is turned on
  - Active threads show dynamic and static SQL

Main Menu Option H



Near-Term History Options

Cannot make changes from this panel, have to edit member COPTxxxx in the library listed below

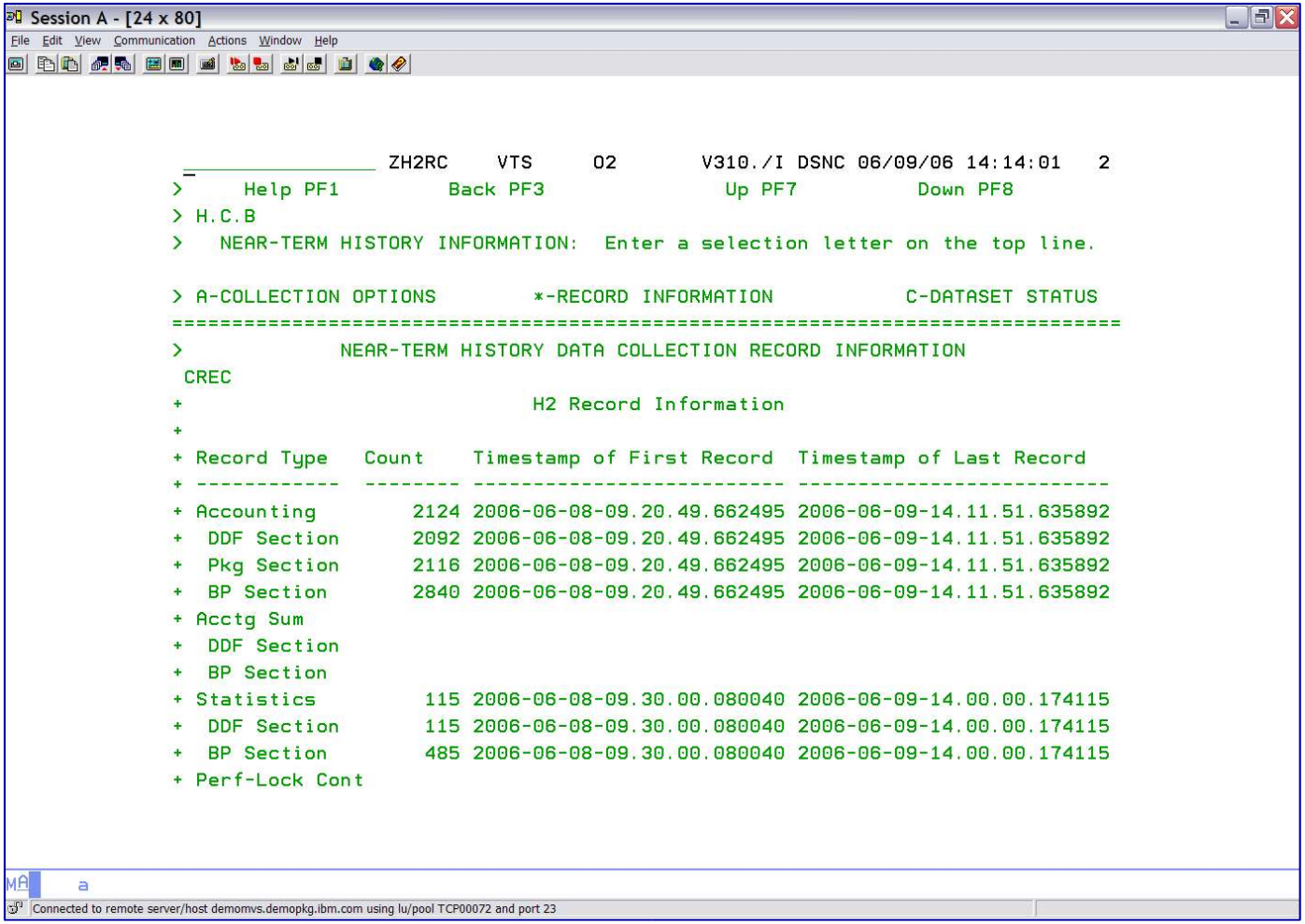


SCAN
DYNAMIC SQL
NEGSQL = NO
LOCKCOUNT
LOCKSUSP

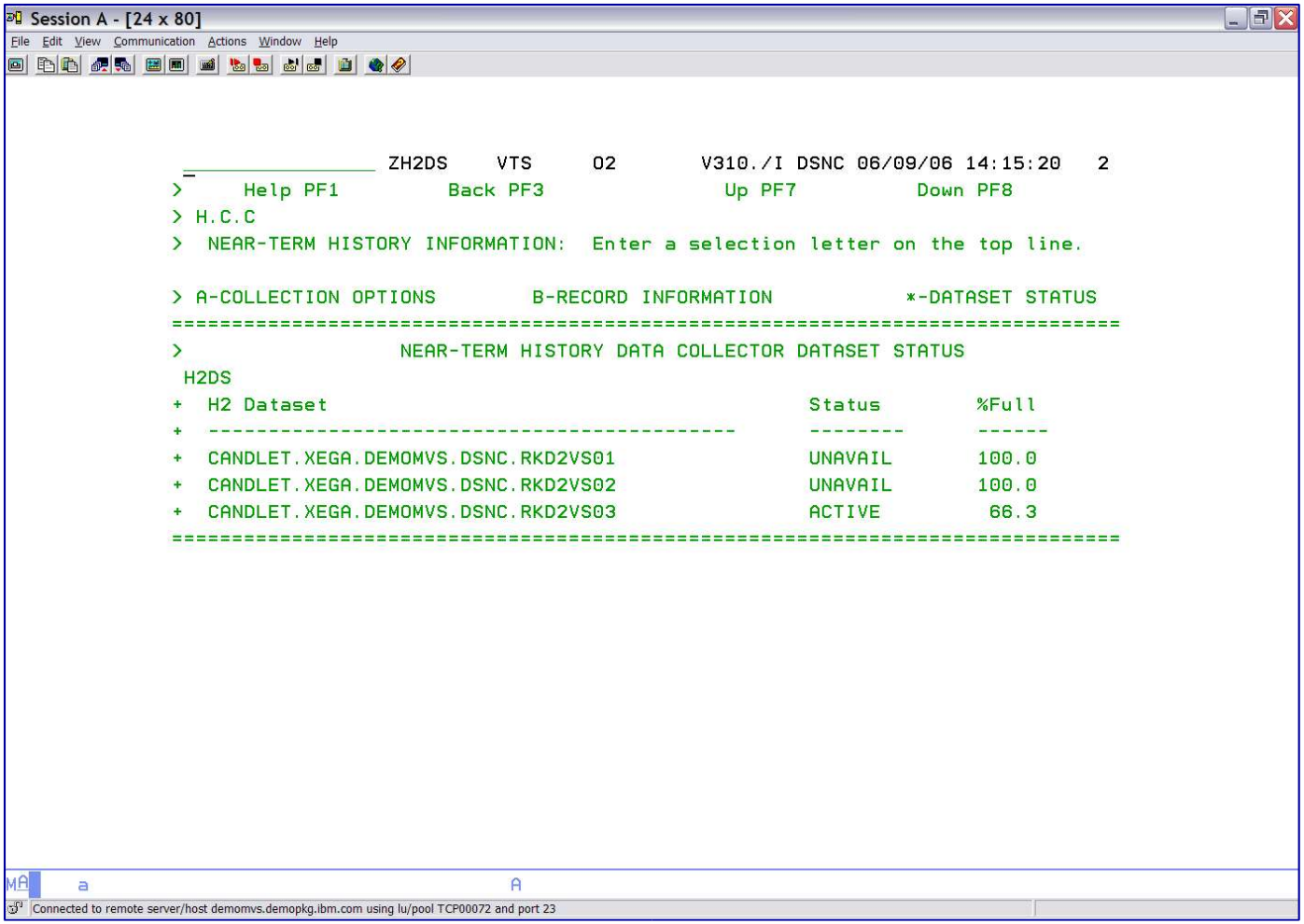
Options are stored in COPTxxxx which resides in this dataset

/F H2xxxx,VARY OPTION=COPTxxxx

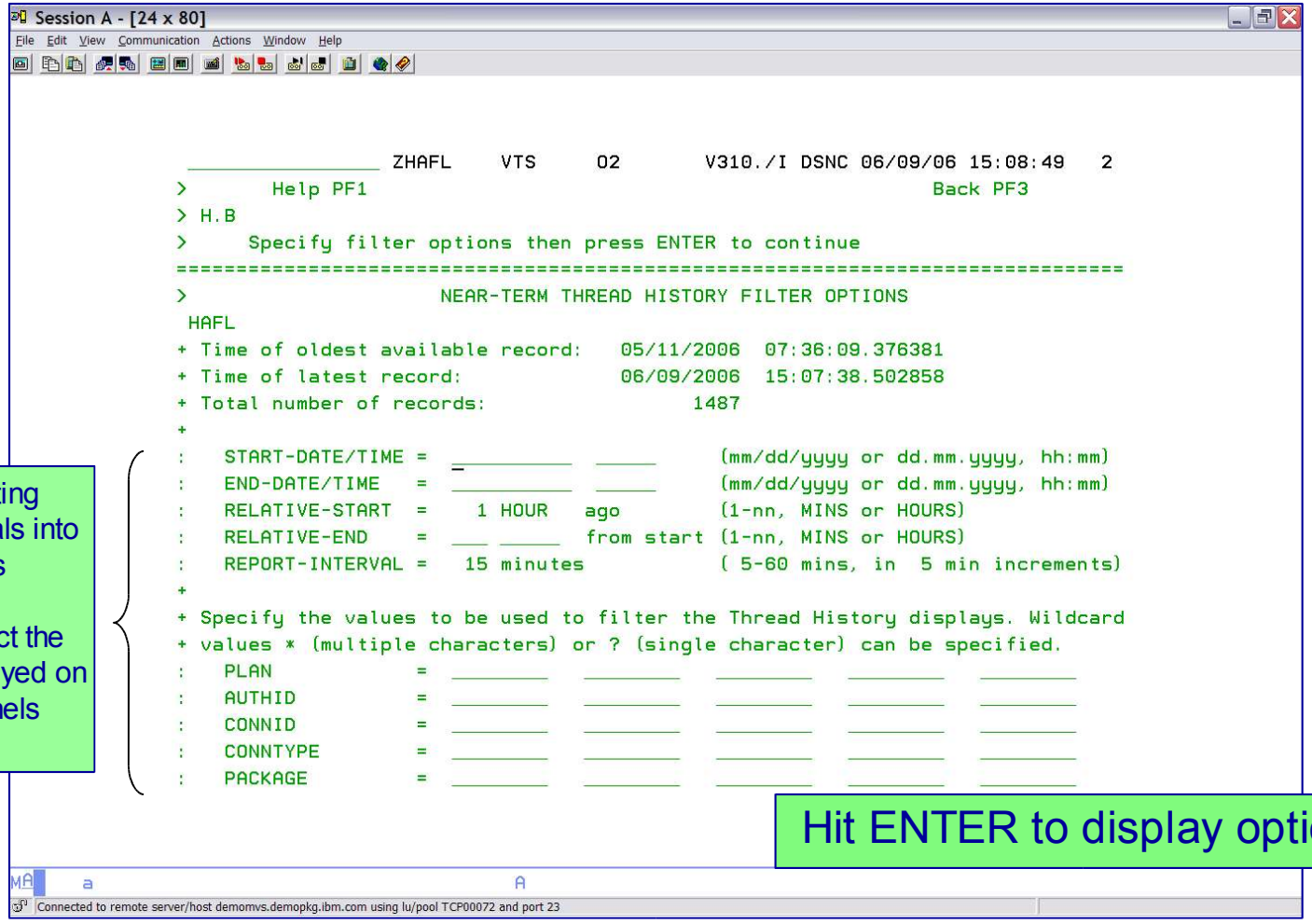
Near-Term History Collection Summary



Near-Term History Data Set Status



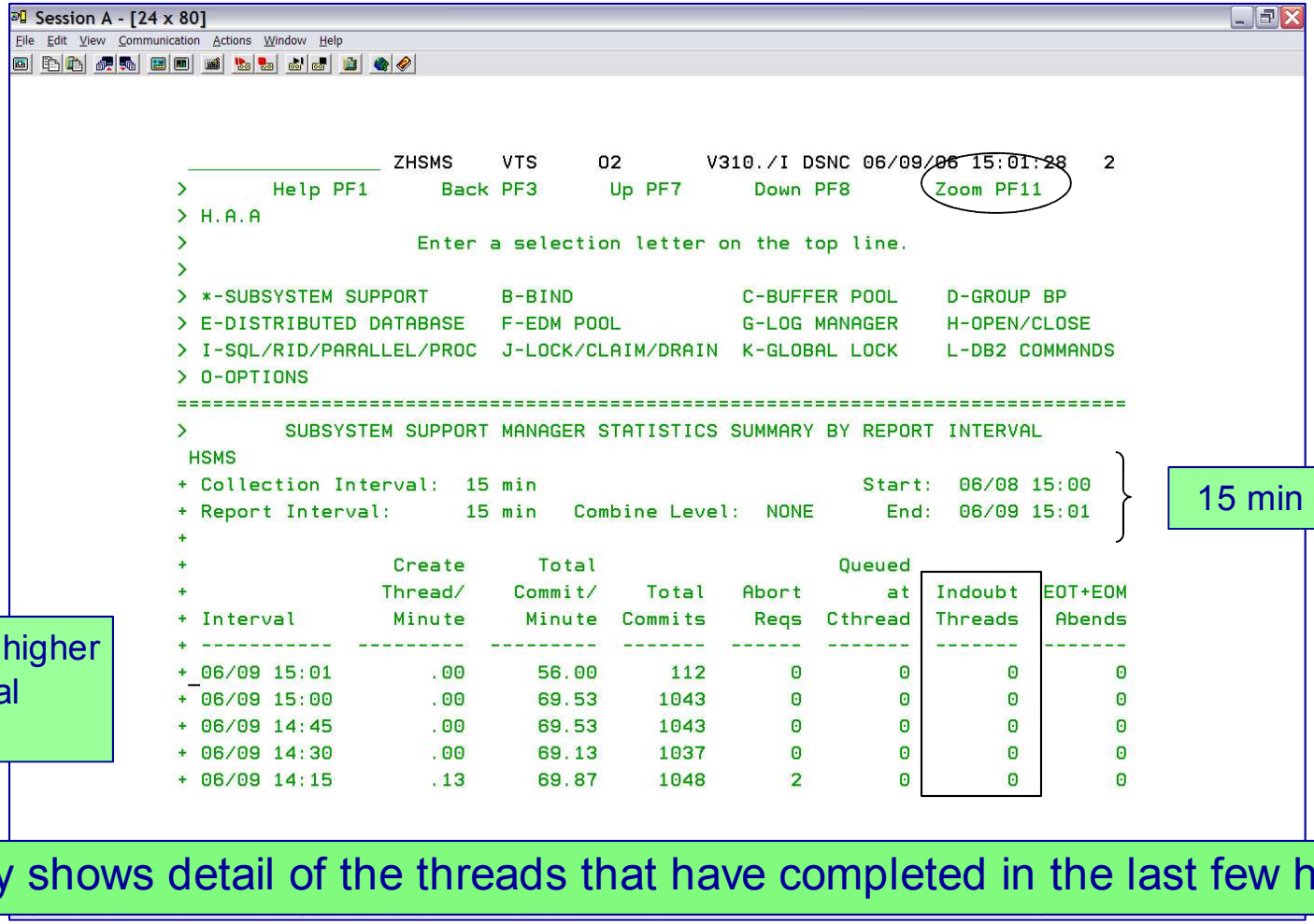
Near-Term History Filter Options



Specify the reporting period, the intervals into which the report is divided, and filter options that restrict the data that is displayed on thread history panels

Hit ENTER to display options

Near-Term History Statistics Summary

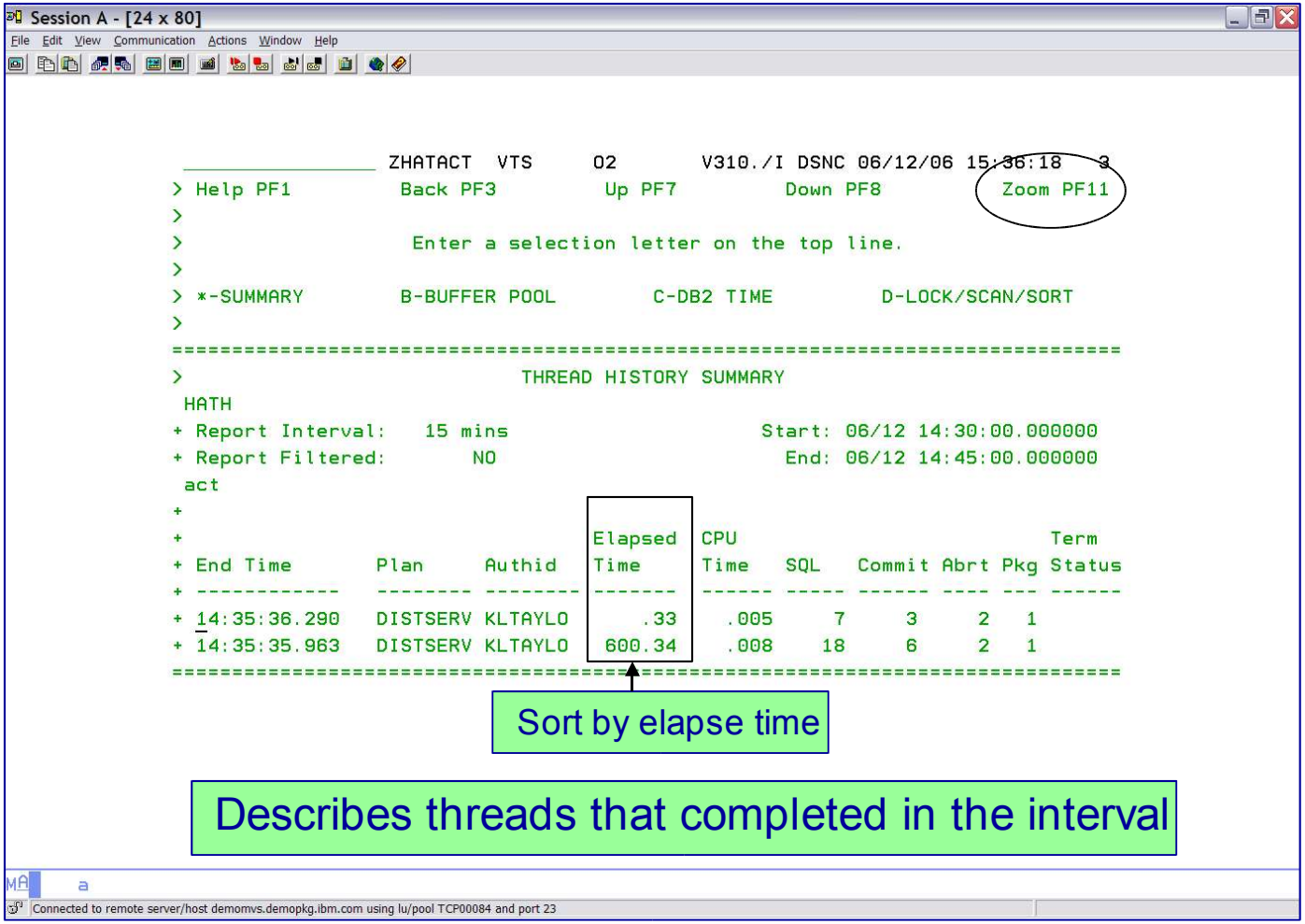


15 min interval

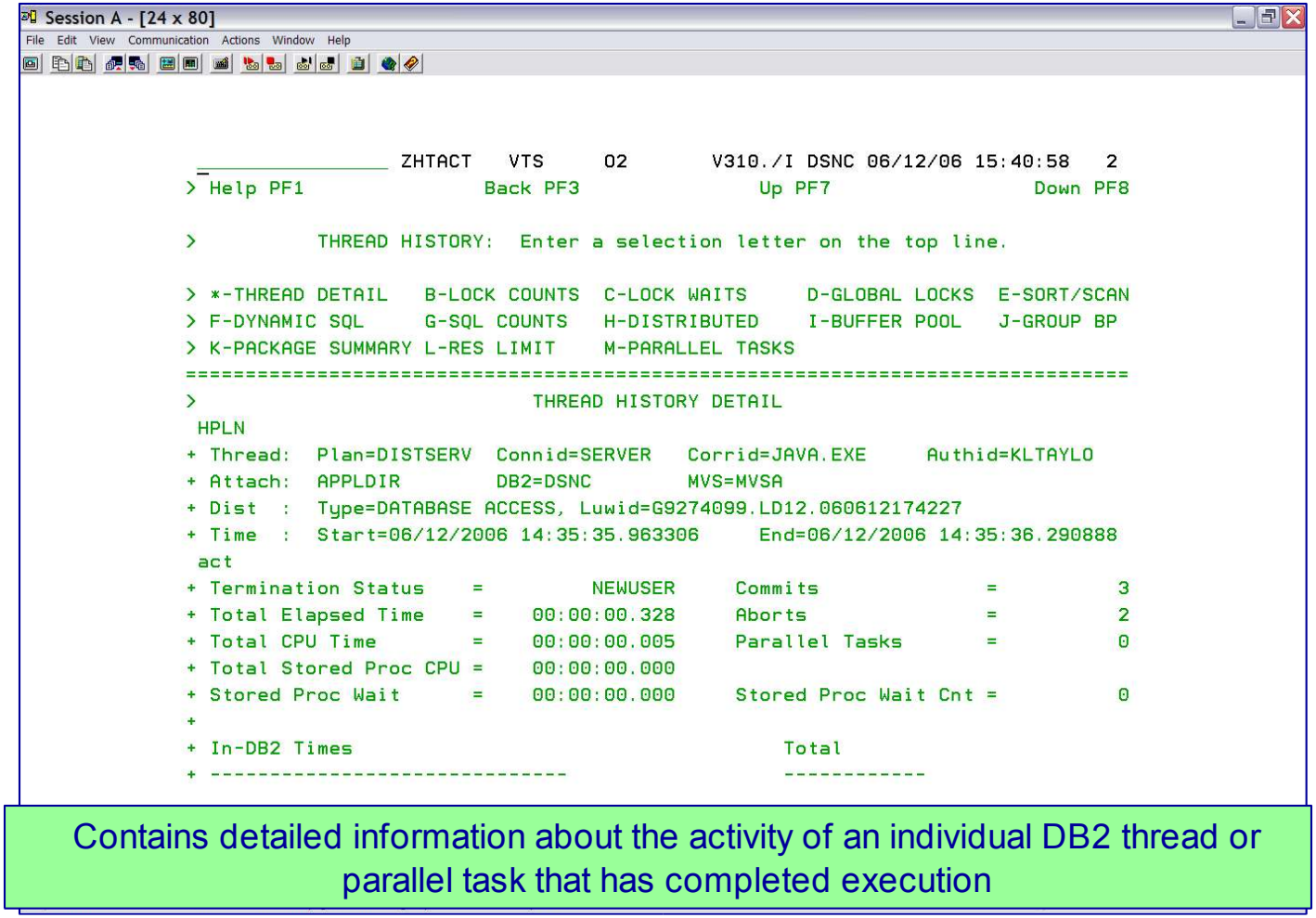
Look for higher than usual numbers

Only shows detail of the threads that have completed in the last few hours

Near-Term History Thread Summary



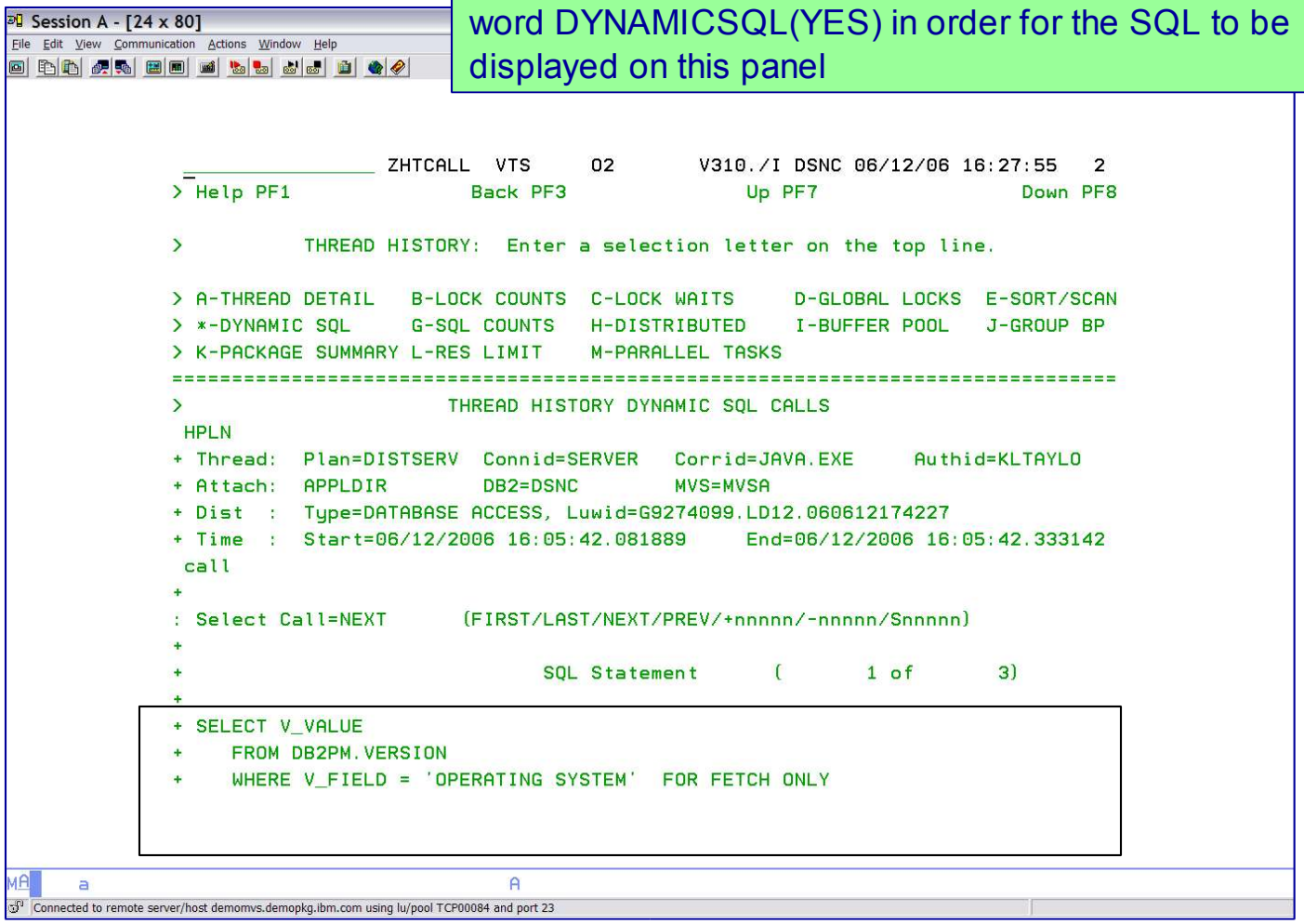
Near-Term History Thread Detail



Contains detailed information about the activity of an individual DB2 thread or parallel task that has completed execution

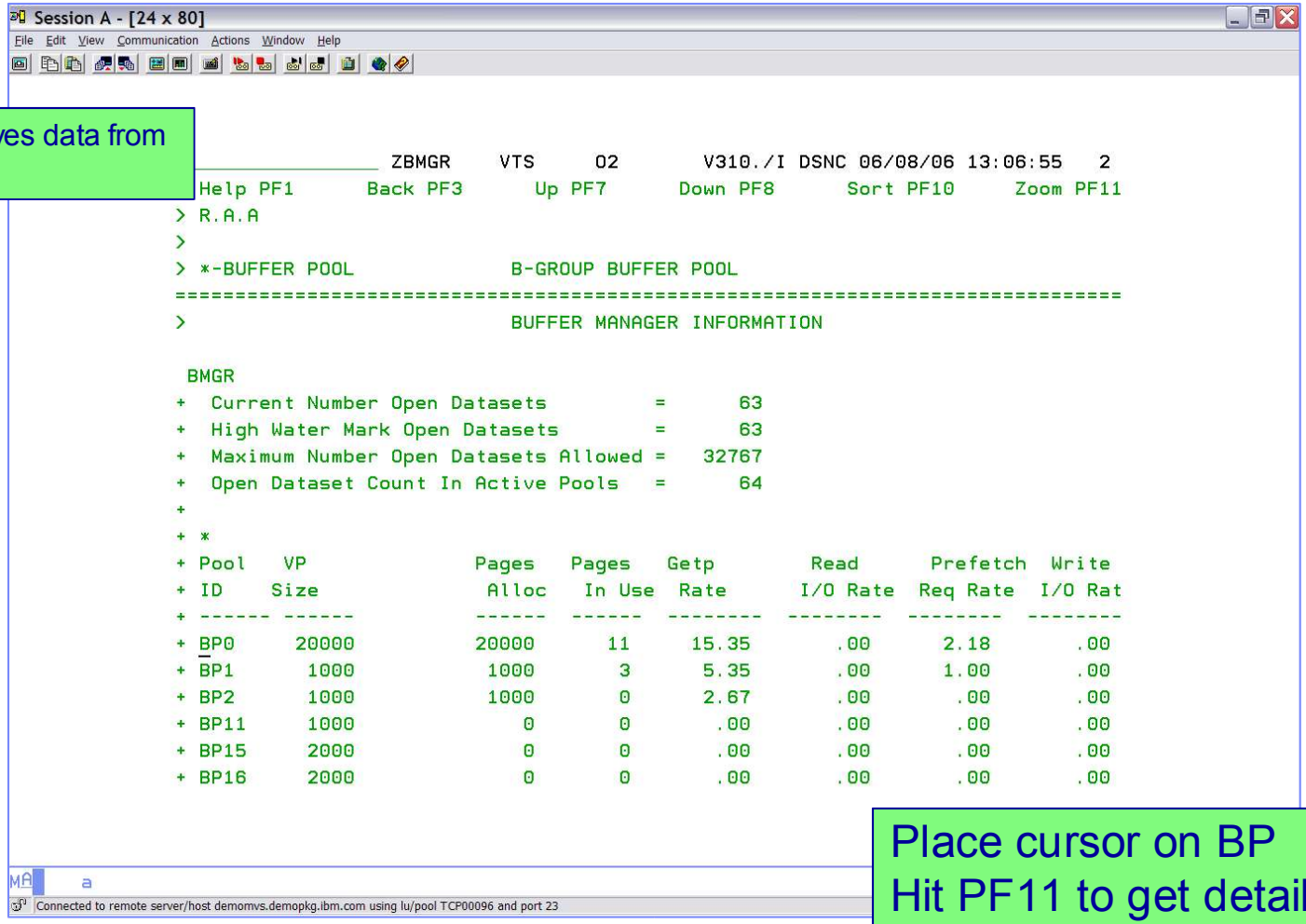
Near-Term History Dynamic SQL

When starting the online historical collector, you must use the word DYNAMICSQL(YES) in order for the SQL to be displayed on this panel



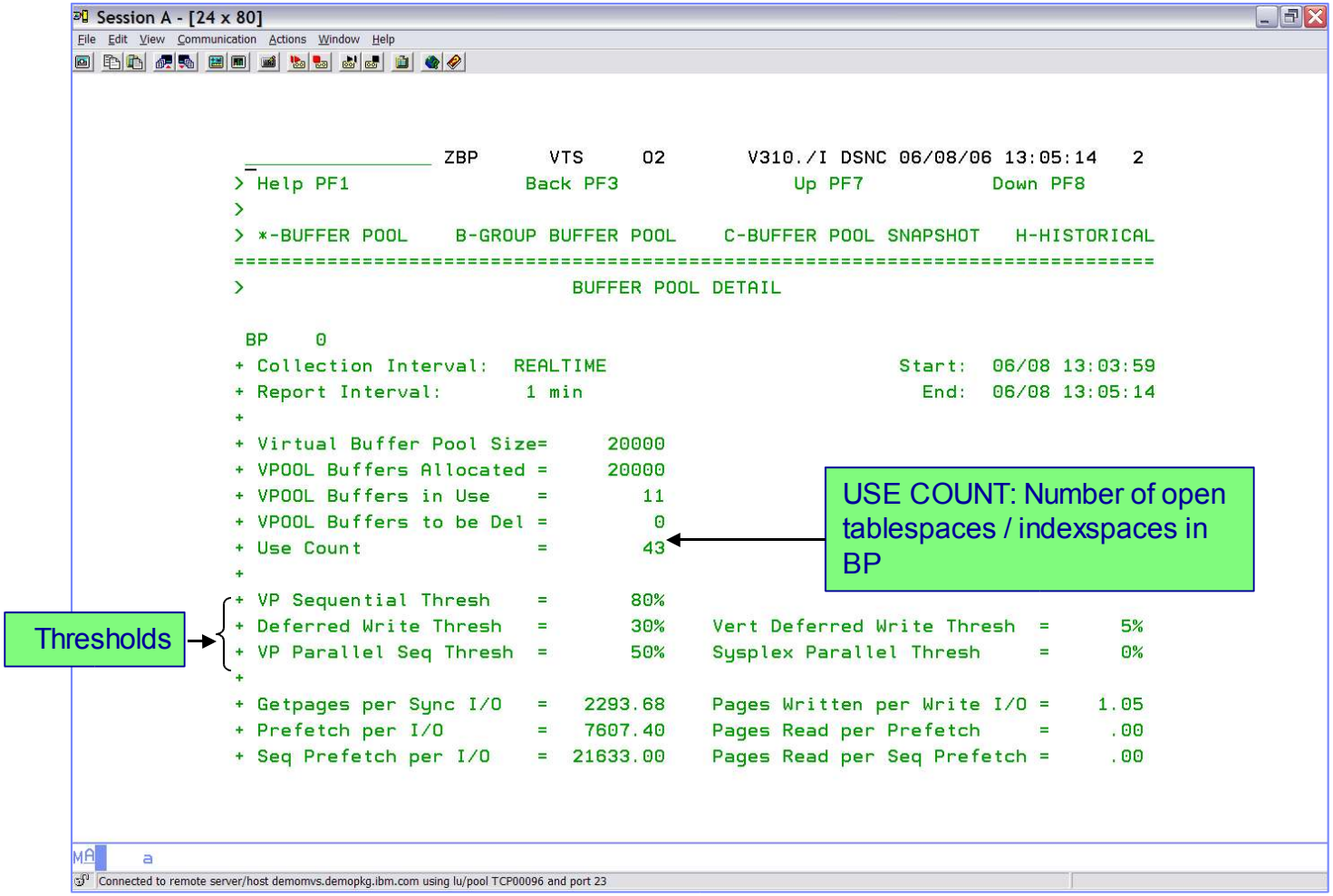
Buffer Manager Information

Buffer Manager moves data from disk to memory



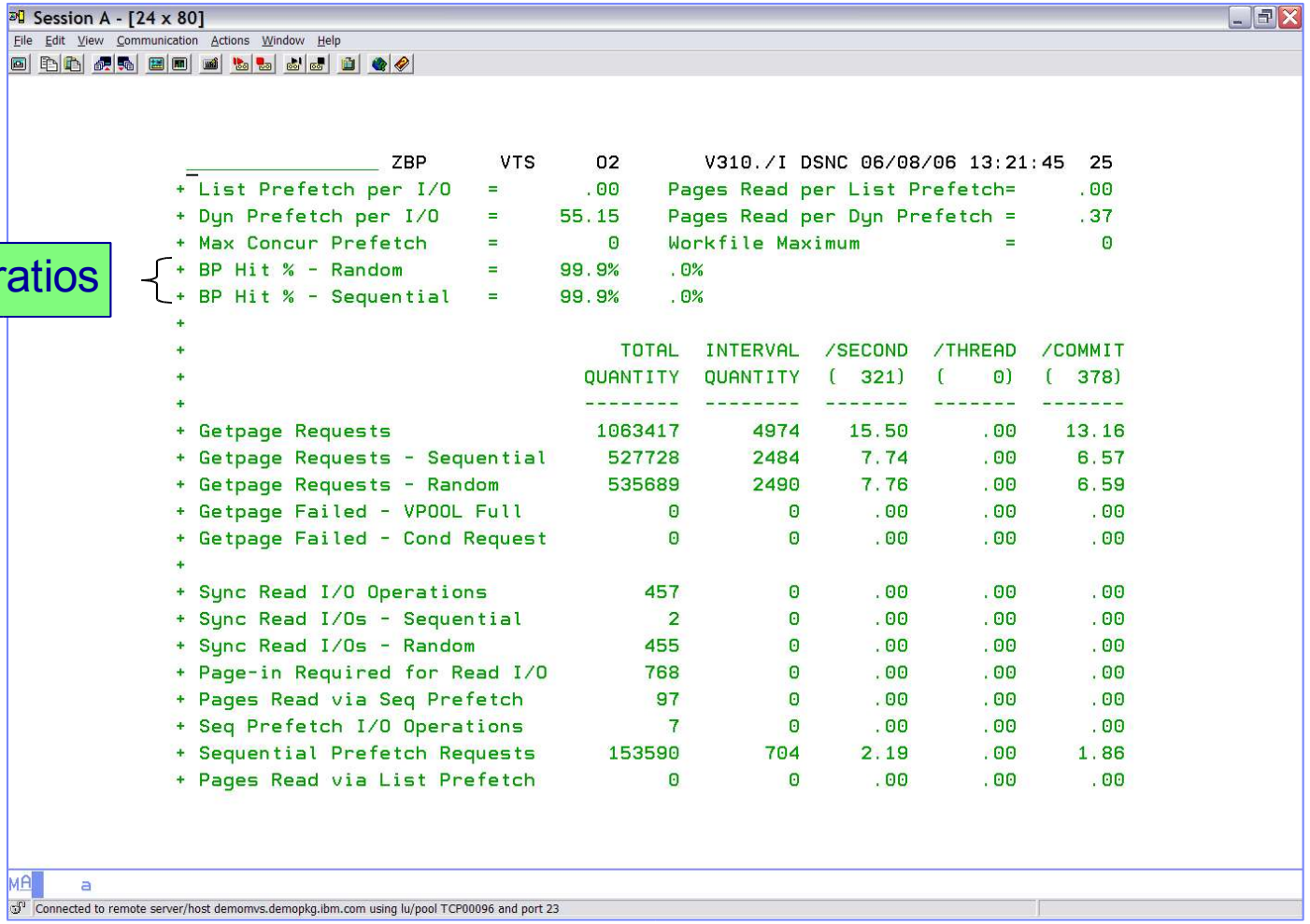
Place cursor on BP
Hit PF11 to get details

Buffer Pool Detail



Buffer Pool Detail

Hit ratios



Session A - [24 x 80]

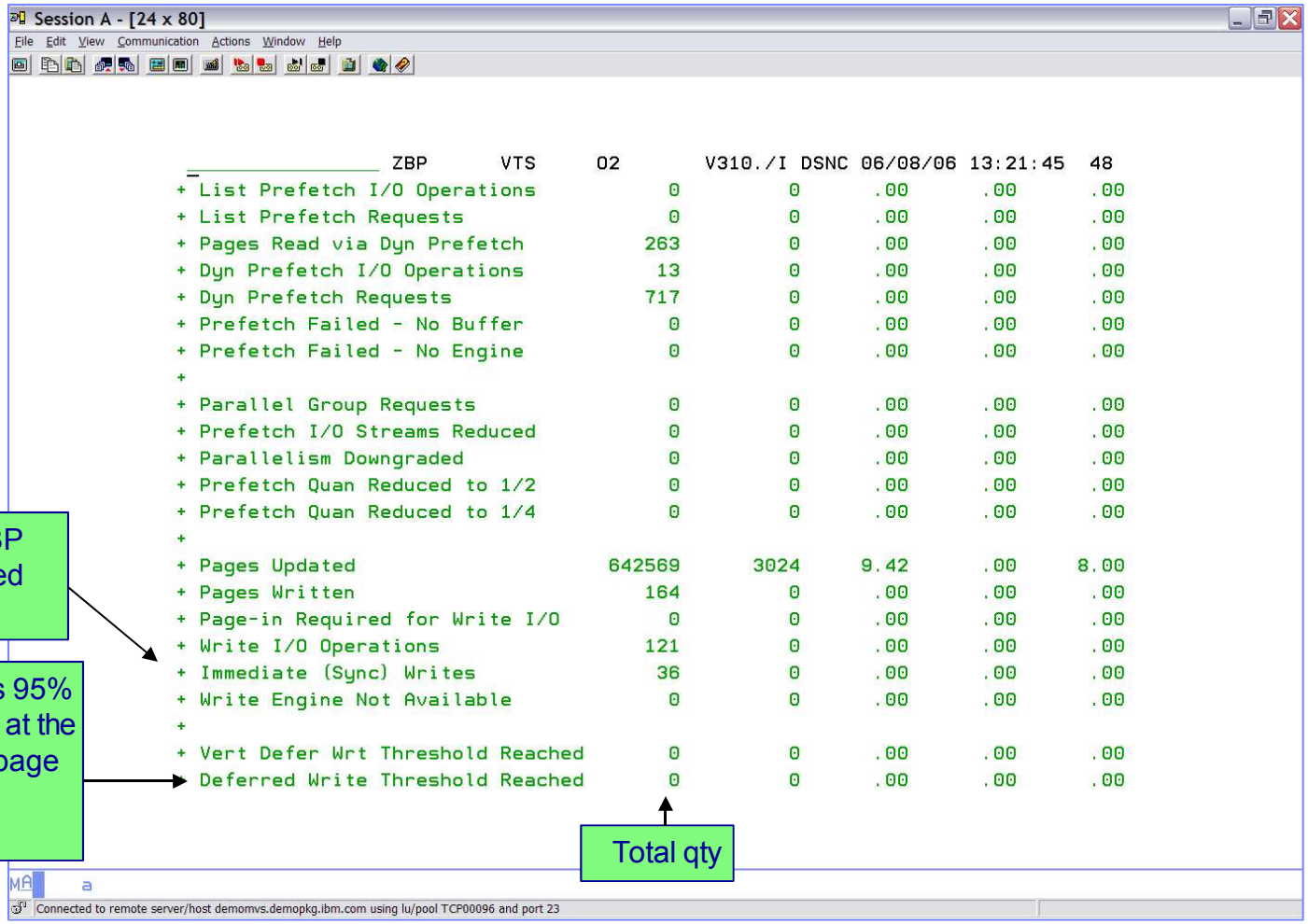
File Edit View Communication Actions Window Help

```
_________________ ZBP      VTS      02       V310./I DSNC 06/08/06 13:21:45  25
+ List Prefetch per I/O   =       .00    Pages Read per List Prefetch=      .00
+ Dyn Prefetch per I/O    =     55.15    Pages Read per Dyn Prefetch =      .37
+ Max Concur Prefetch     =         0    Workfile Maximum            =         0
+ BP Hit % - Random       =     99.9%    .0%
+ BP Hit % - Sequential   =     99.9%    .0%
+
+                                  TOTAL   INTERVAL  /SECOND  /THREAD  /COMMIT
+                                QUANTITY  QUANTITY  (  321)  (    0)  (  378)
+                                --------  --------  -------  -------  -------
+ Getpage Requests               1063417      4974    15.50      .00    13.16
+ Getpage Requests - Sequential   527728      2484     7.74      .00     6.57
+ Getpage Requests - Random       535689      2490     7.76      .00     6.59
+ Getpage Failed - VPOOL Full          0         0      .00      .00      .00
+ Getpage Failed - Cond Request        0         0      .00      .00      .00
+
+ Sync Read I/O Operations           457         0      .00      .00      .00
+ Sync Read I/Os - Sequential          2         0      .00      .00      .00
+ Sync Read I/Os - Random            455         0      .00      .00      .00
+ Page-in Required for Read I/O      768         0      .00      .00      .00
+ Pages Read via Seq Prefetch         97         0      .00      .00      .00
+ Seq Prefetch I/O Operations          7         0      .00      .00      .00
+ Sequential Prefetch Requests    153590       704     2.19      .00     1.86
+ Pages Read via List Prefetch         0         0      .00      .00      .00
```

MA a

Connected to remote server/host demomvs.demopkg.ibm.com using lu/pool TCP00096 and port 23

Buffer Pool Detail



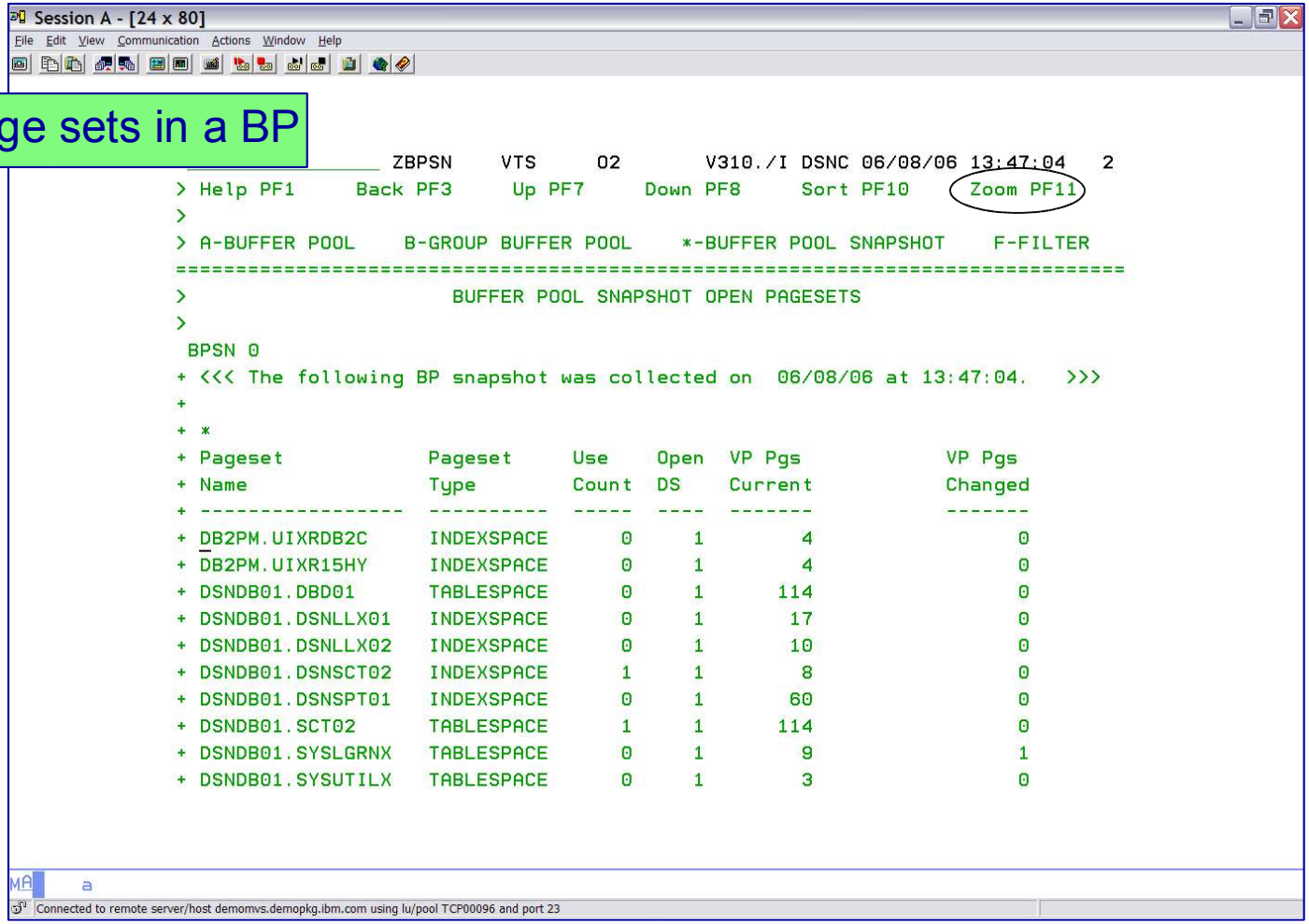
IWTH – when 97.5% of BP pages are in use, deferred write is disabled

DWTH – when DB2 uses 95% of BP, begins to operate at the row level rather than the page level = increased CPU

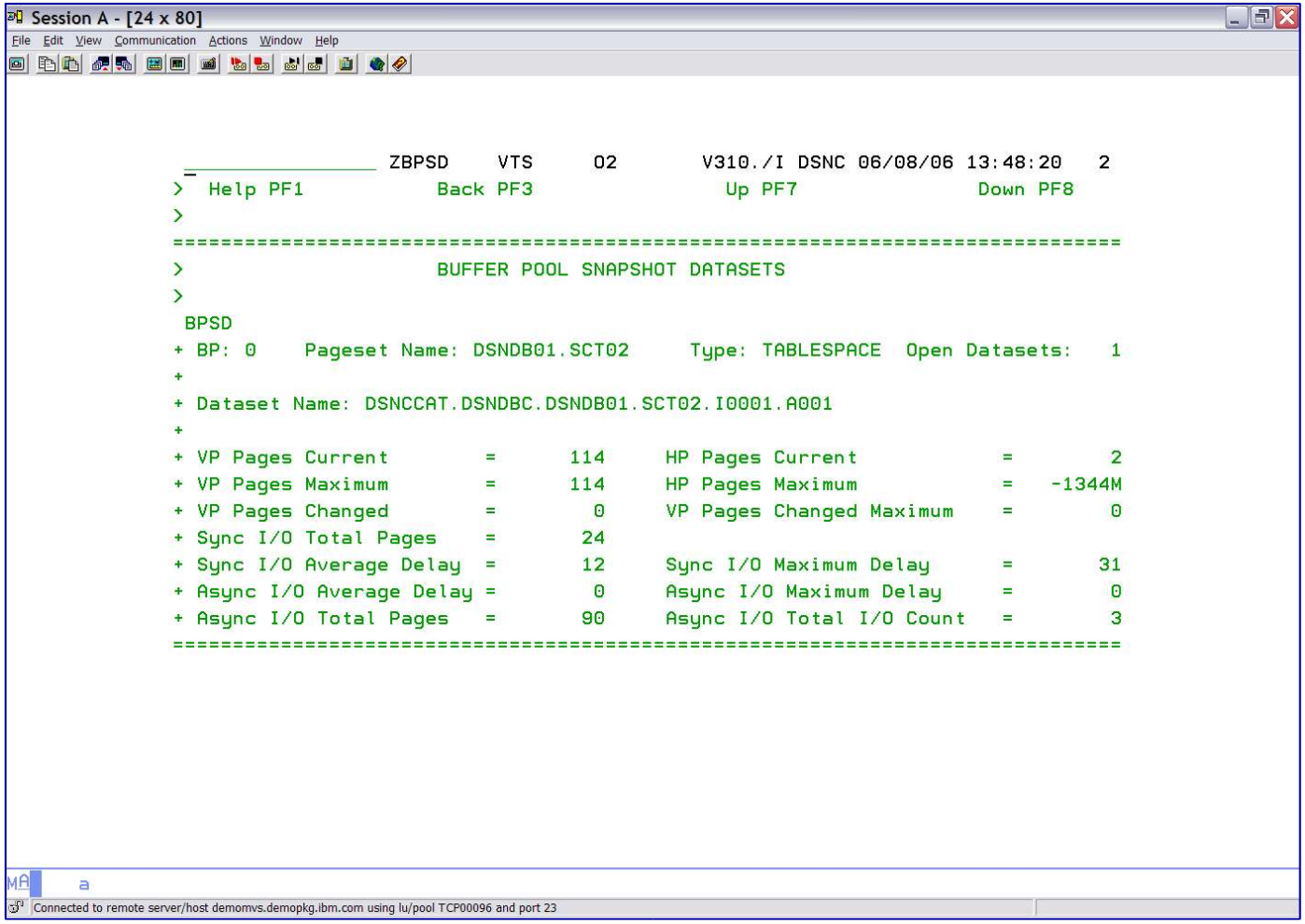
Total qty

Buffer Pool Snapshot

Open page sets in a BP

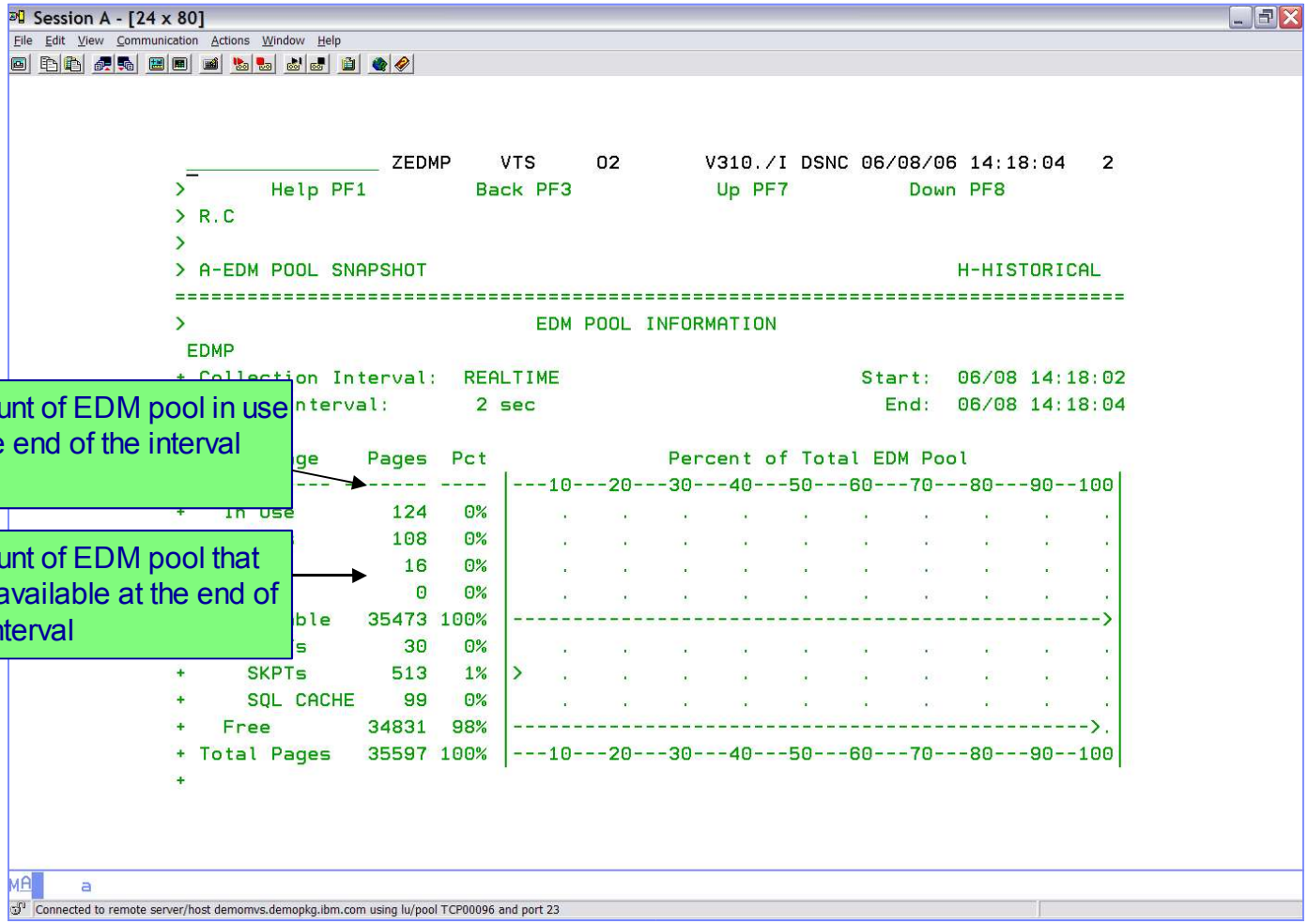


Buffer Pool Snapshot



EDM Pool

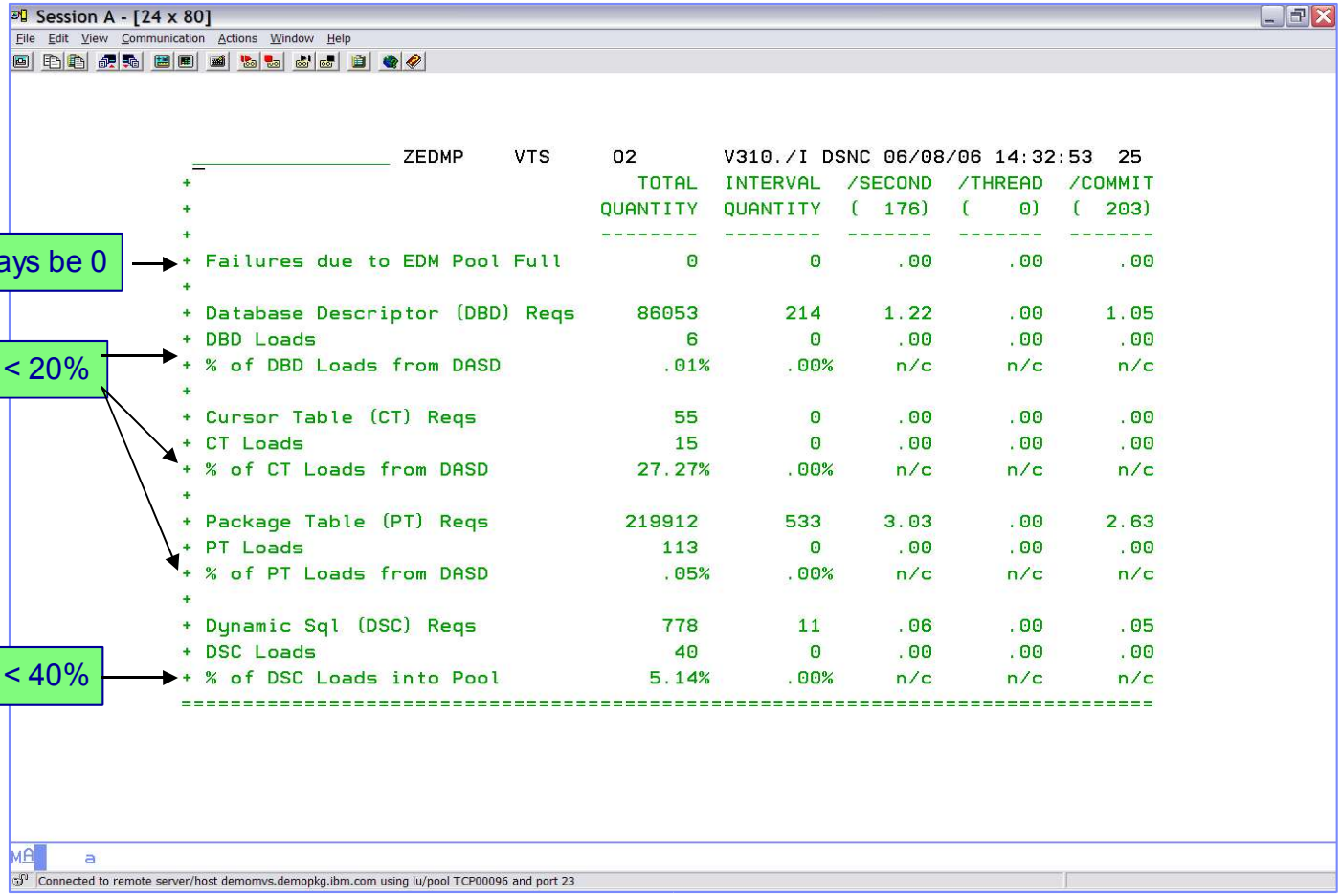
- Used to maintain DBD's, cursor tables, authorization cache, and dynamic statement cache
- Size specified in DSNZPARM



Amount of EDM pool in use at the end of the interval

Amount of EDM pool that was available at the end of the interval

EDM Pool

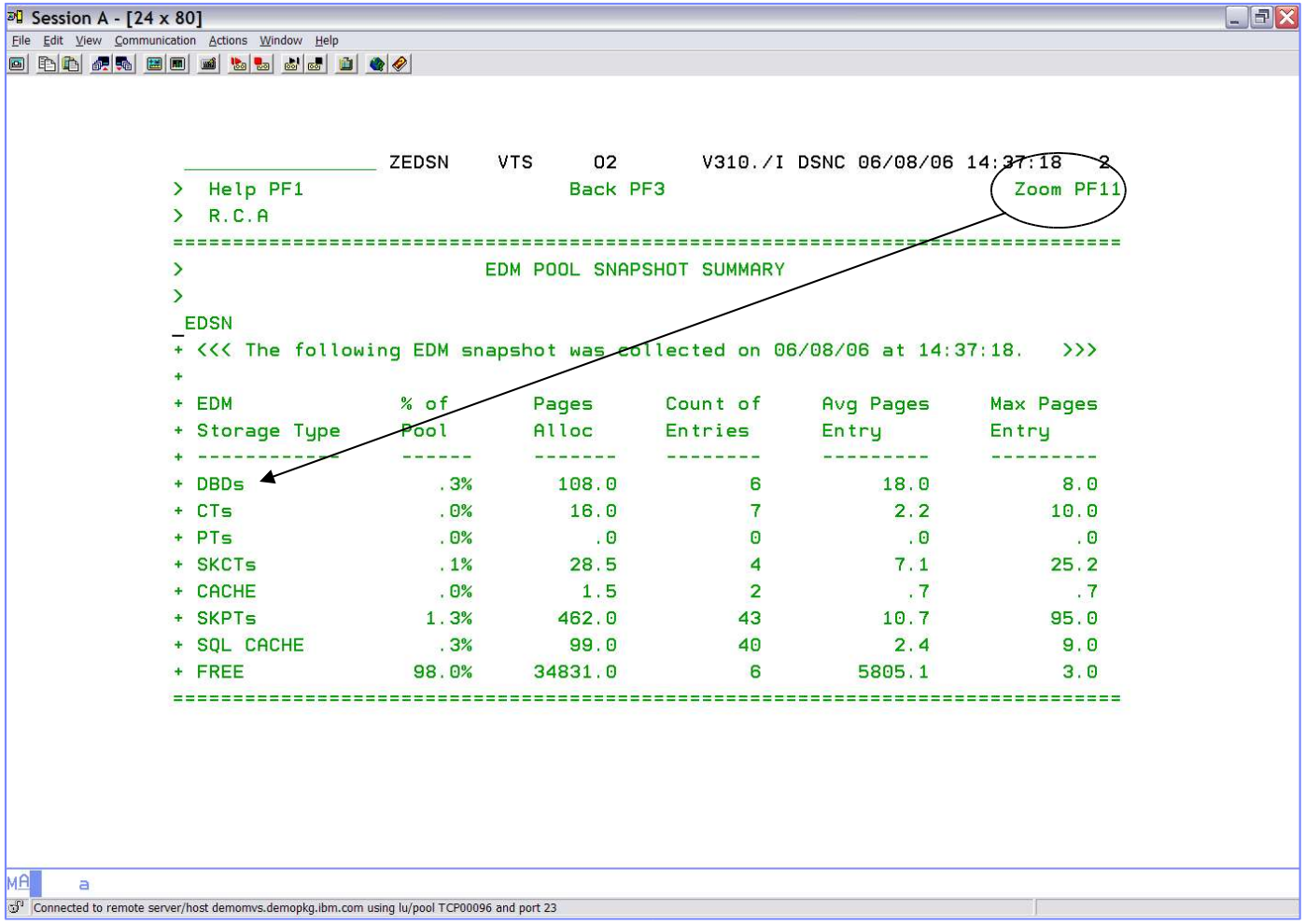


Should always be 0

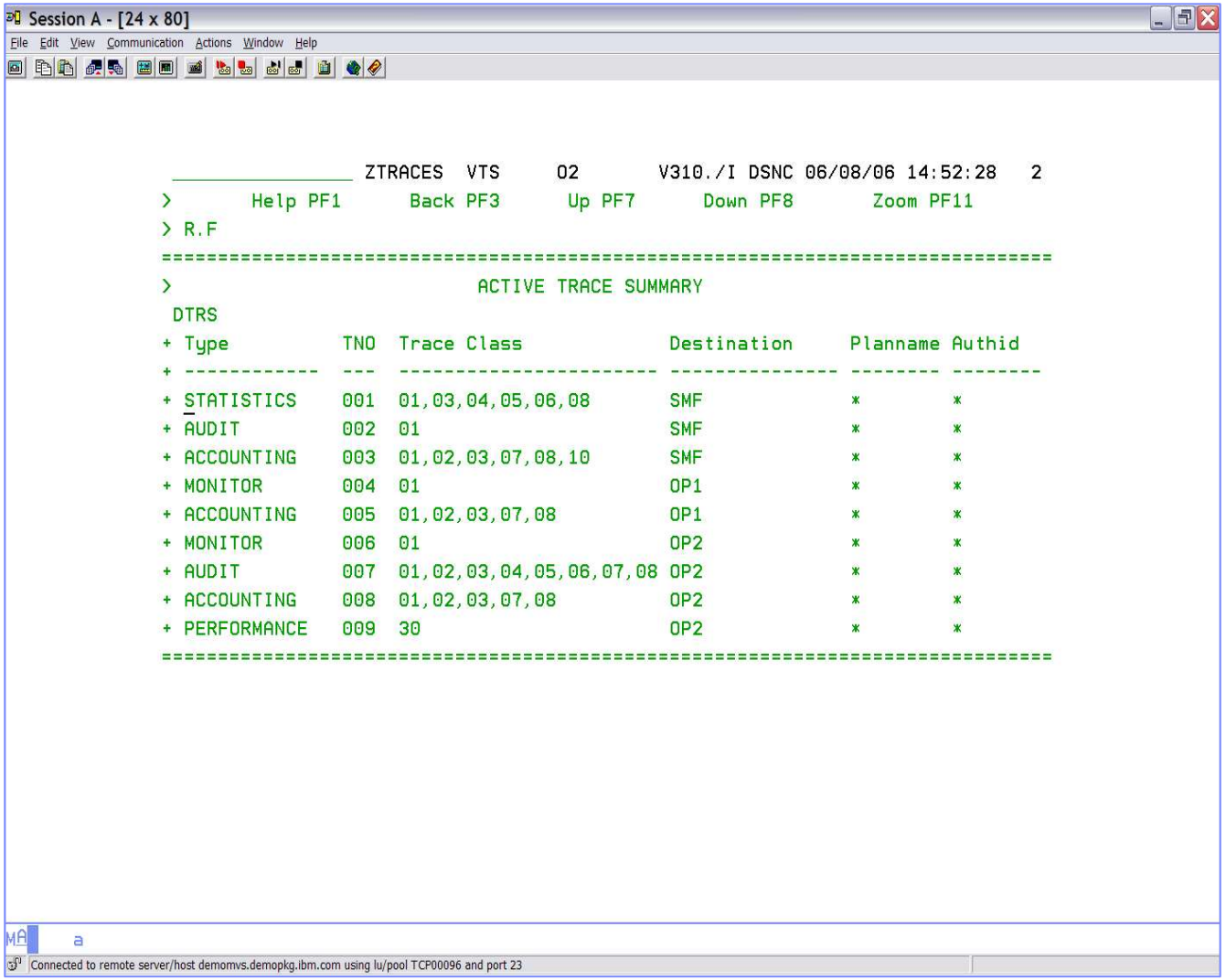
Should be < 20%

Should be < 40%

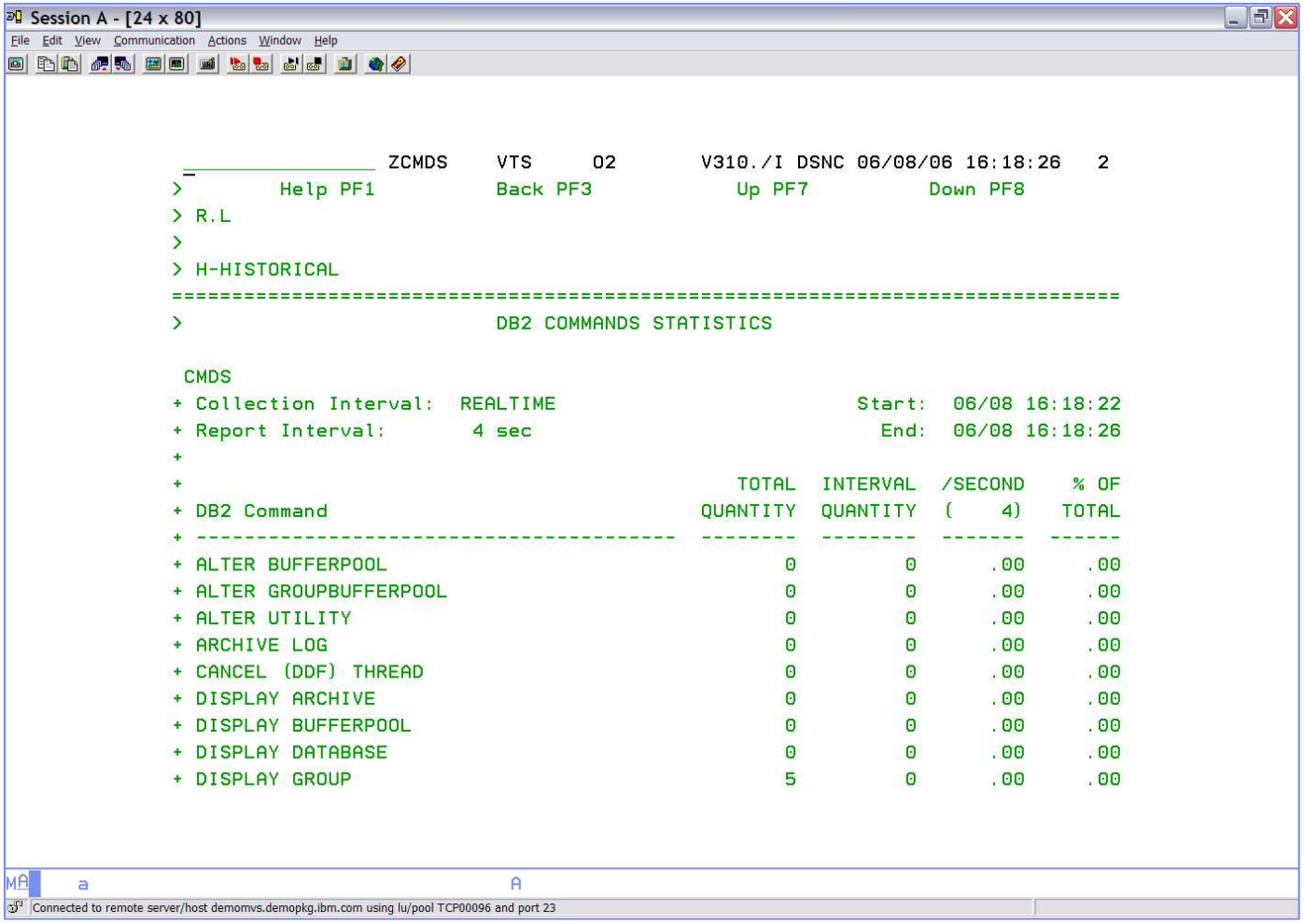
EDM Pool Snapshot



Active Traces



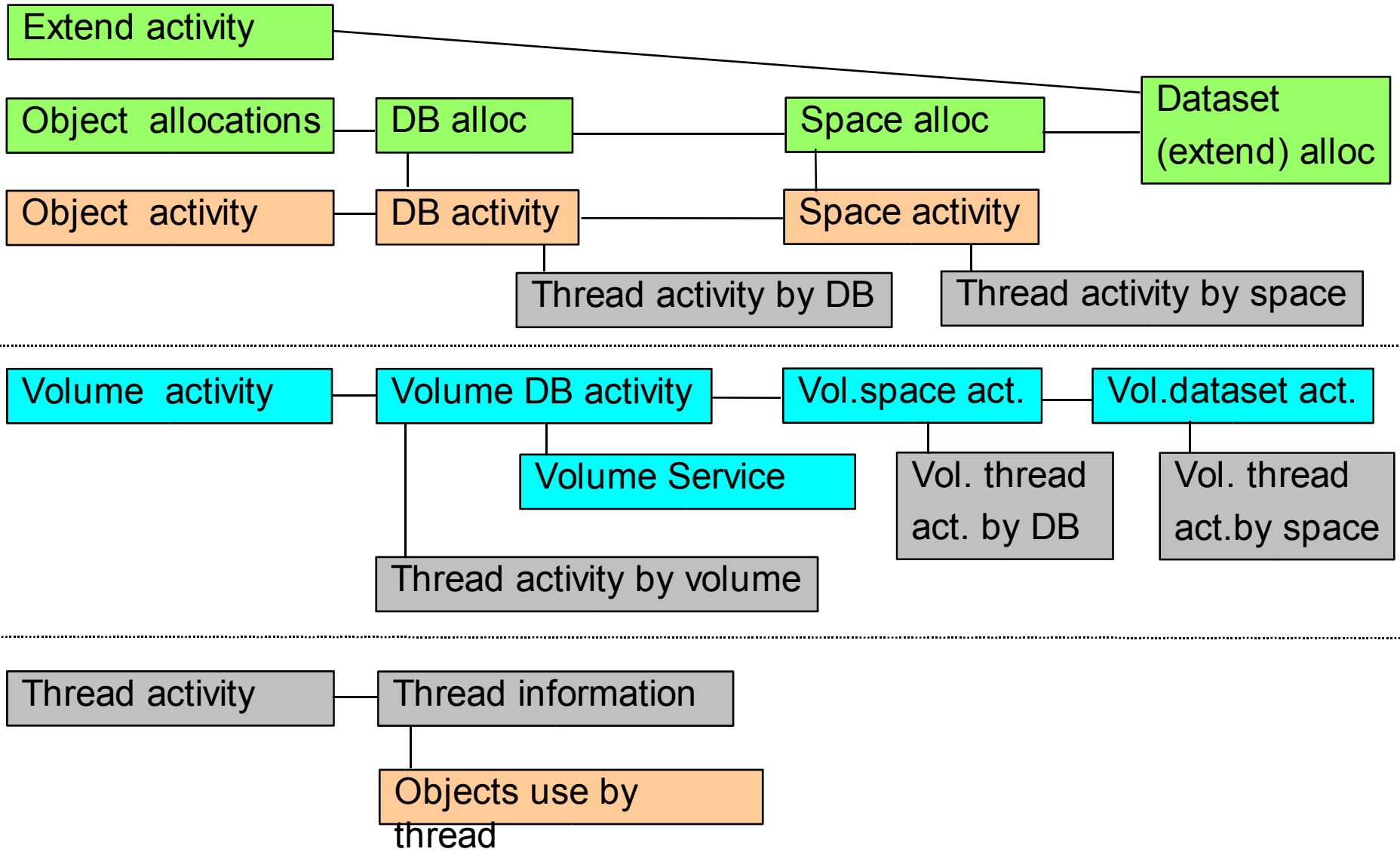
DB2 Commands



Object Analysis

- Analyzes objects that are currently allocated
- Best used for reactive problems - data is not stored
- Provides information regarding
  - Linear dataset extend activity
  - I/O activity at the DASD volume level
  - Object activity from a DB2 perspective
- Does not use traces
- Overhead is in OMPE server started task

Object Analysis



Object Analysis

- How to start
  - ICAT - AUTOSTART function for Object / Volume analysis - would be enabled when the OMPE server is activated (not recommended)
  - Manually - START OBJECT ANALYSIS COLLECTORS Panel (O.F from the Main Menu)
    - Have to select an interval expressed in minutes (1 – 1440)
    - EVENTMGR has to be activated
      - Issue the MVS Modify command from SDSF console interface

        F *name.of.ompe.server.address.space,*DISPLAY
      - Must be active – if not issue the following command

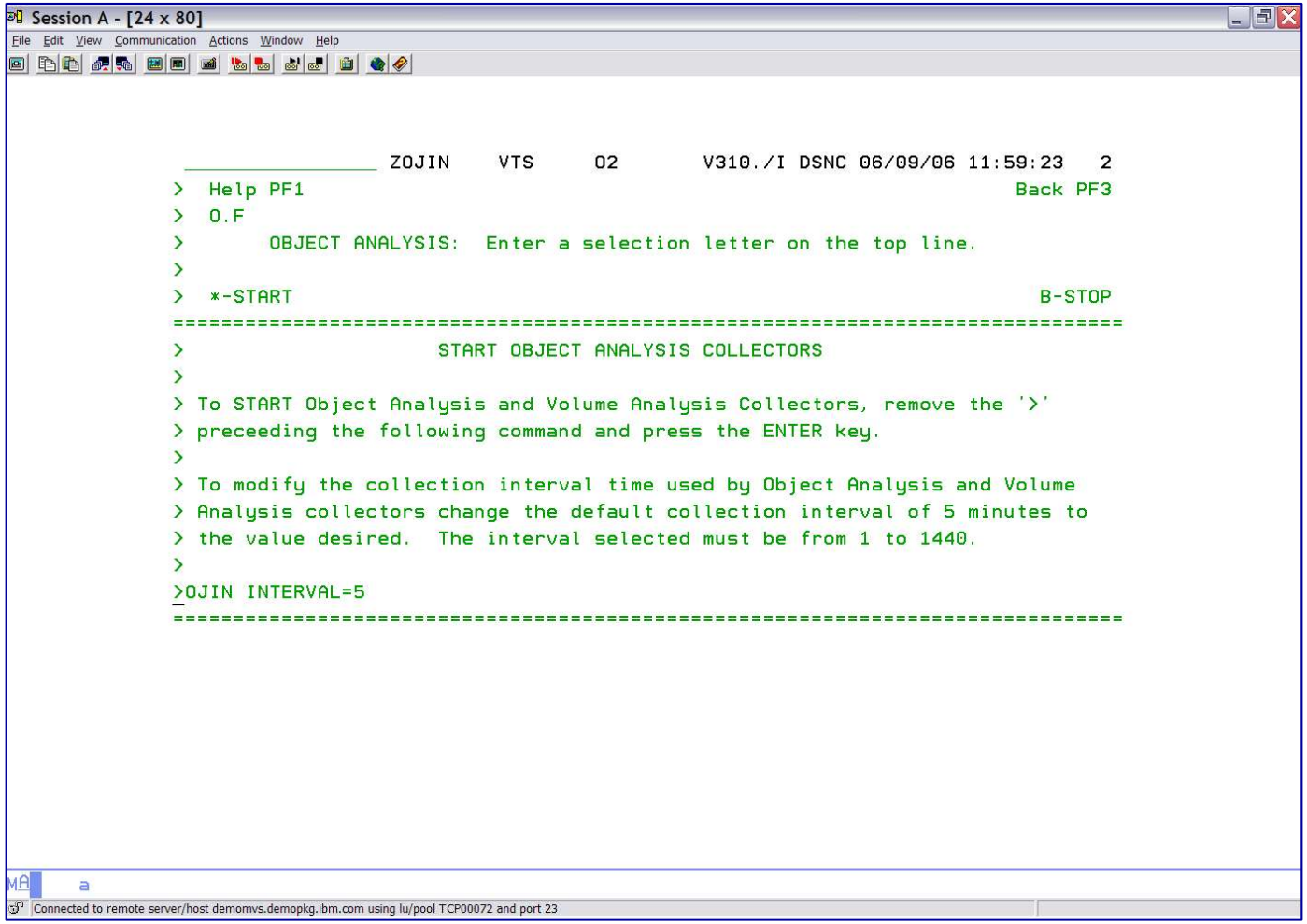
        F *name.of.ompe.server.address.space* S EVENTMGR

Recommendation: Automatically start EVENTMGR and manually start Object Analysis without thread level detail

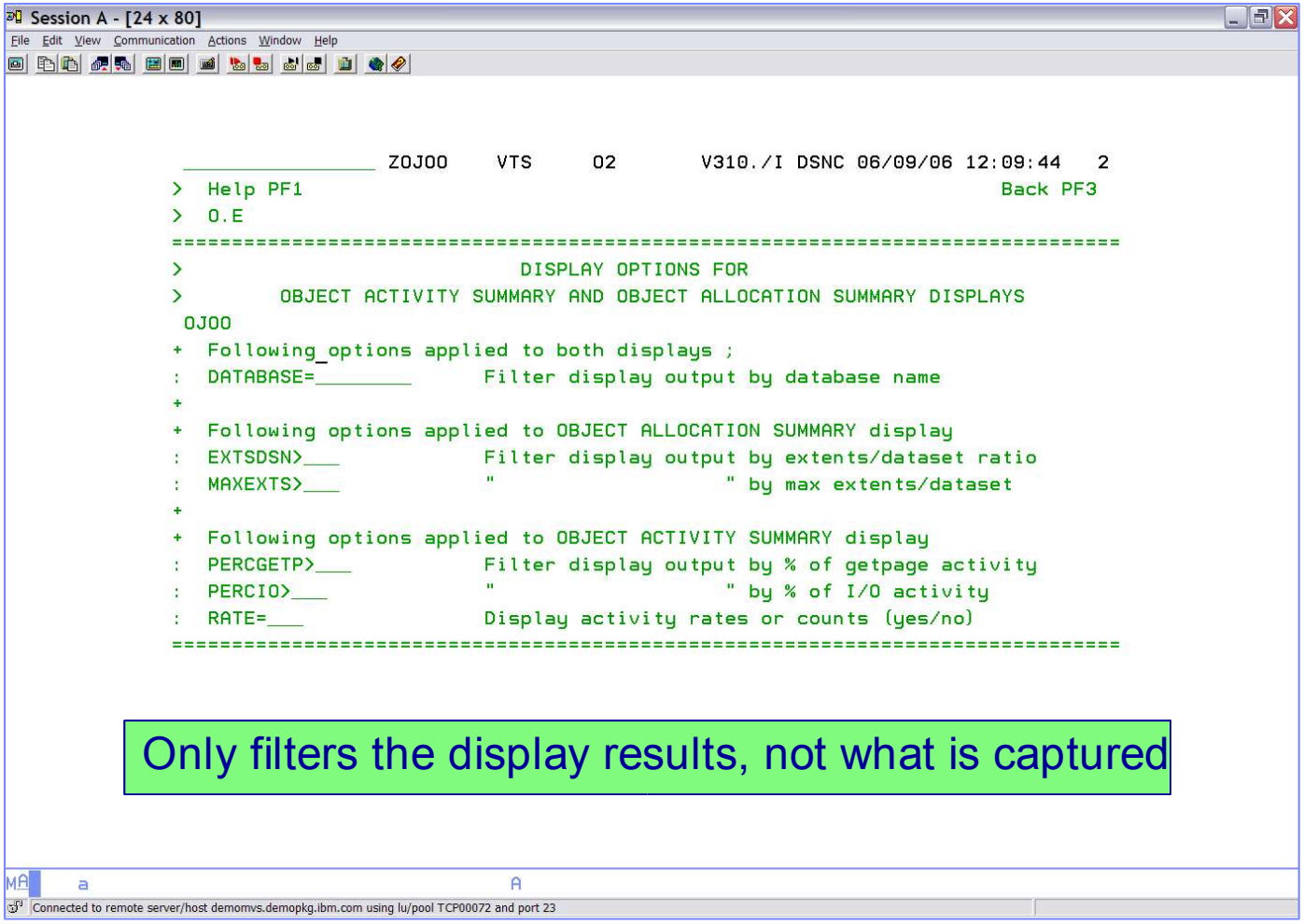
F *ompe.server.address.space,* F EVENTMGR,START DB2=*ssid,* INTERVAL=nn,THREAD=NO

Manually Start Object Analysis

- Main Menu Option O - Object Analysis
- Option F - Collector Control

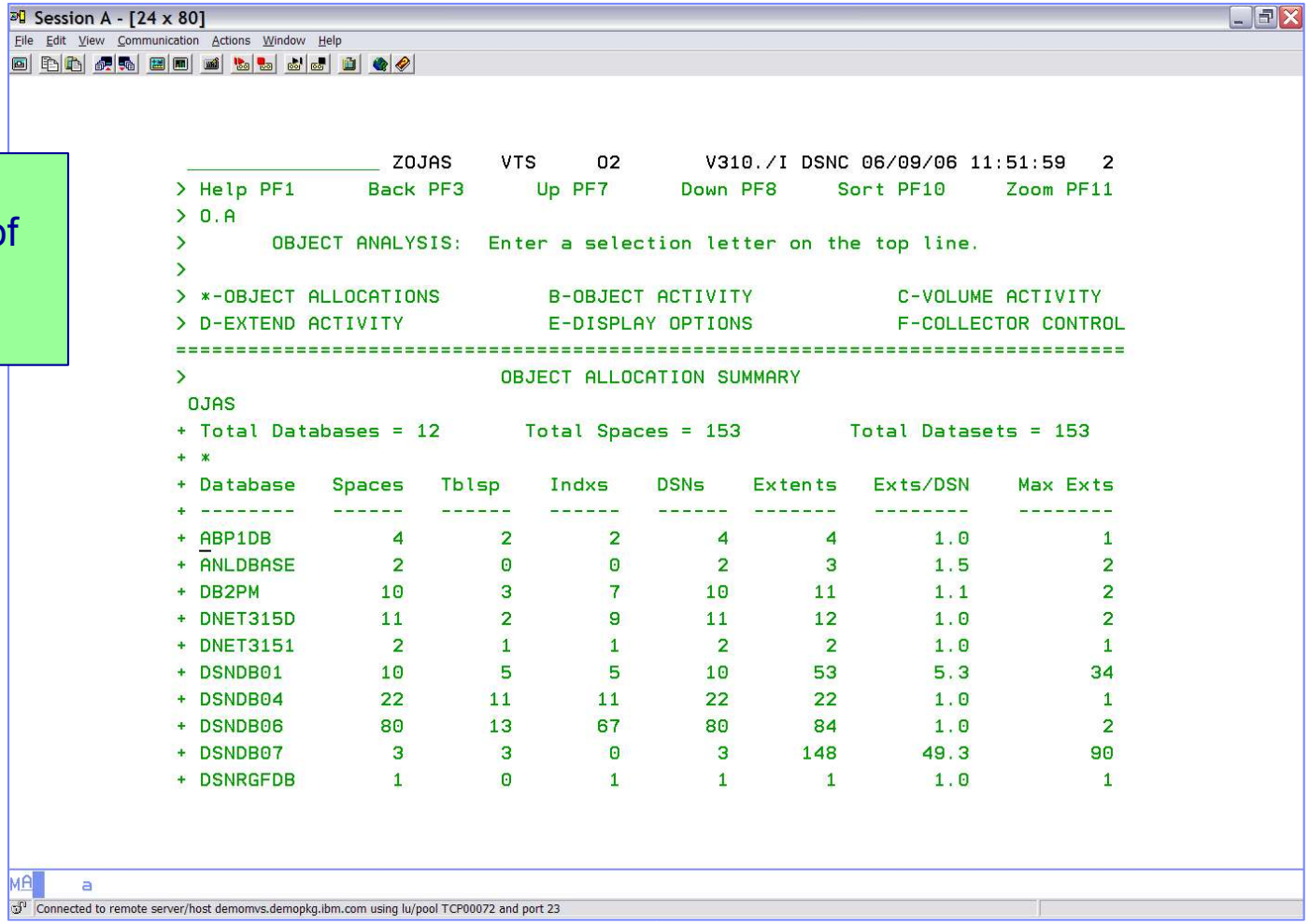


Filtering Objects

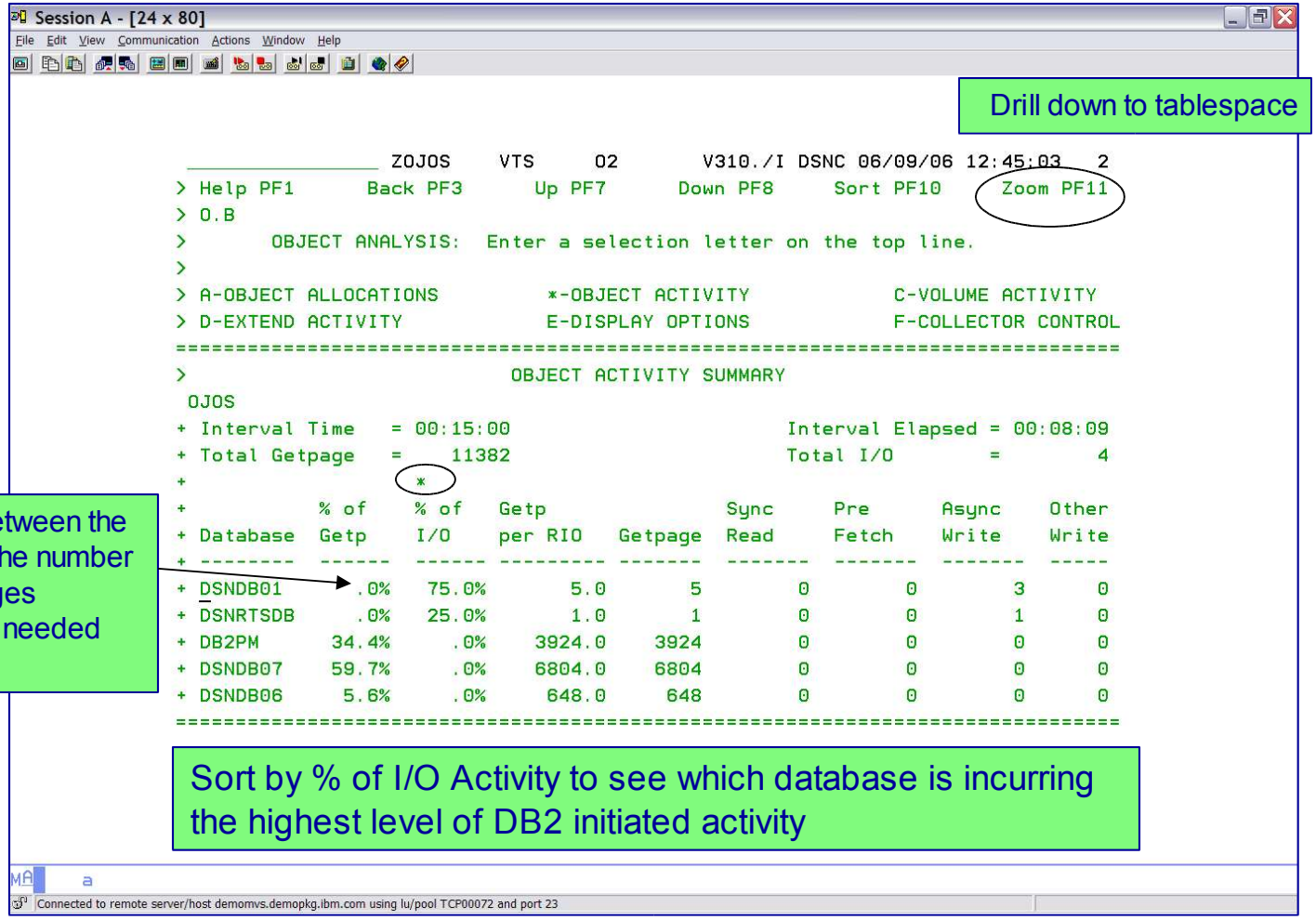


Object Allocations

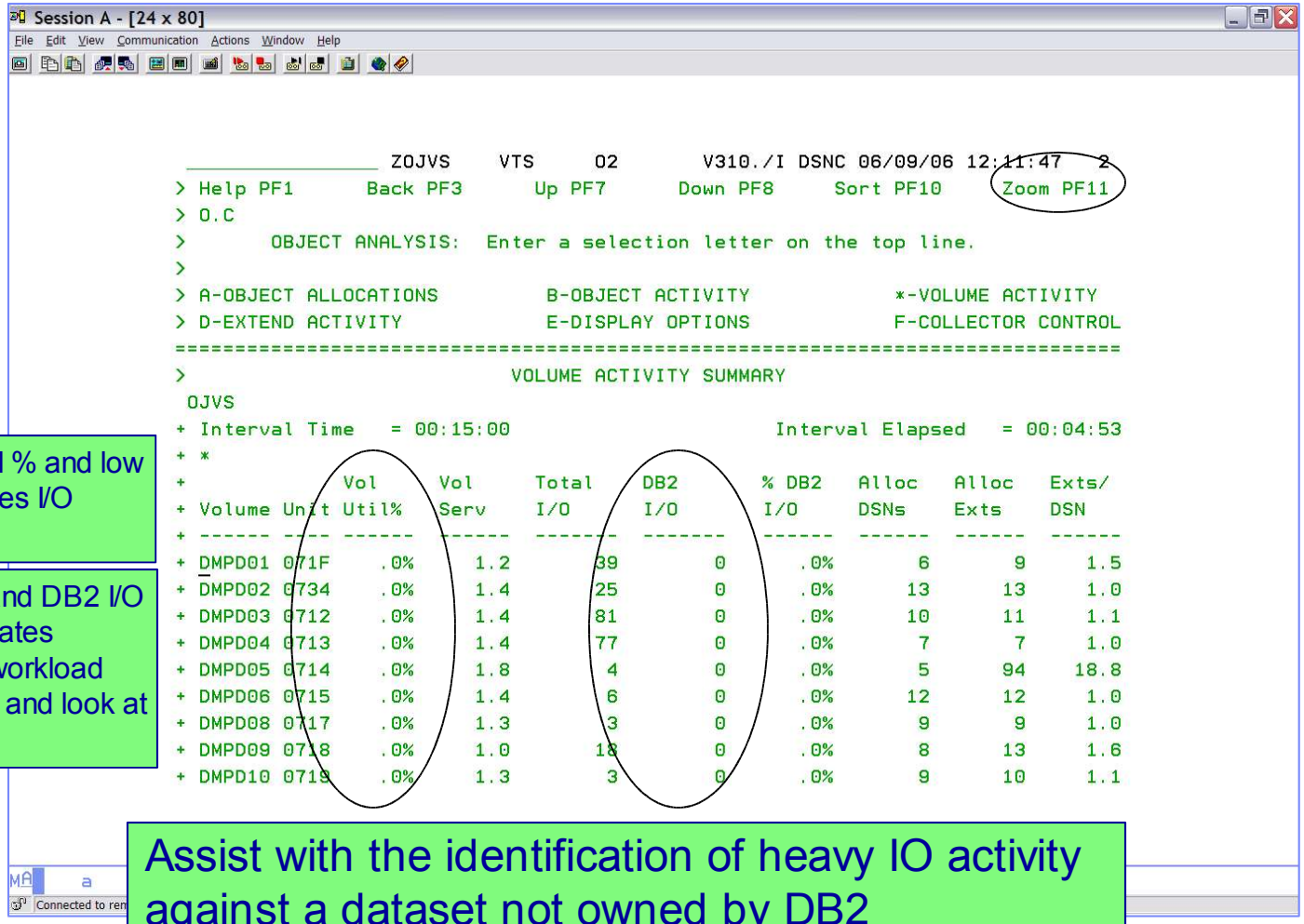
Display allocations of allocated databases



Objects I/O Activity



Volume Activity



High volume util % and low DB2 I/O indicates I/O outside of DB2

Volume util % and DB2 I/O are close, indicates potential DB2 workload issue – may go and look at object activity

Assist with the identification of heavy IO activity against a dataset not owned by DB2

Documentation

- SC18-9988 Monitoring Performance from the OMEGAMON Classic Interface Version 4.1.0
  - How to use the 3270 Classic Interface
- SC18-9981 Monitoring Performance from ISPF Version 4.1.0
  - How to use the PE ISPF Interface
- SC18-9982 Monitoring Performance from Performance Expert Client Version 4.1.0
  - How to use the PE Workstation Client

OMPE Batch Reports

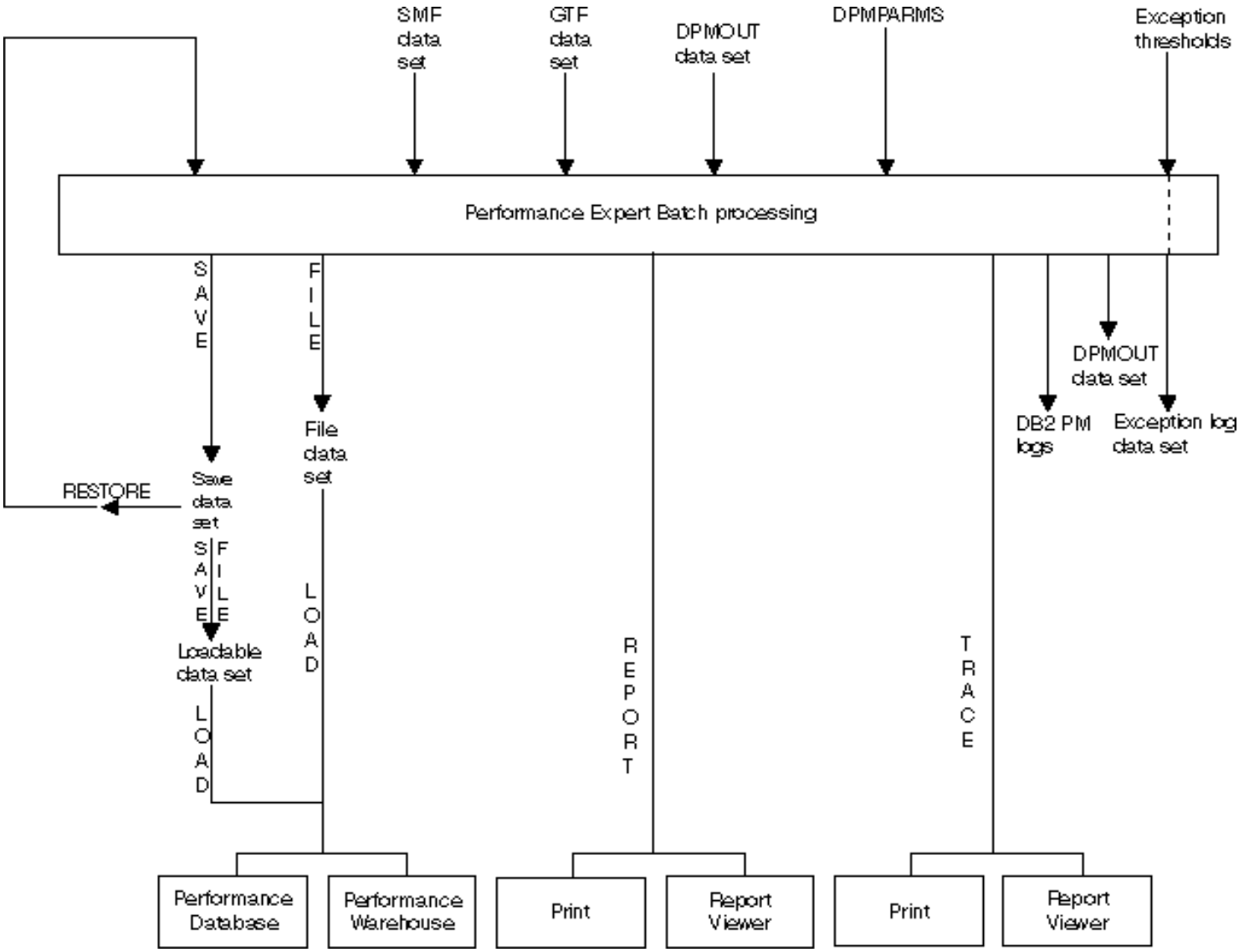
IBM Software Group

ON DEMAND BUSINESS™

Basics

- Batch reporting is based on terminated DB2 activities
  - These activities are reflected in previously collected DB2 trace records
  - Information is provided at various levels of detail depending on need
  - Filtering allows minimizing the input and the amount of output
  - Ordering allows the ordering of reports by keys and intervals
- Use batch reports to
  - Determine DB2 subsystem performance and efficiency
  - Tune the DB2 subsystem
  - Identify and resolve performance problems
  - Measure an application's performance and resource cost
  - Tune applications and SQL queries

Data Flow



Input

- DB2 instrumentation created by selected traces
  - SMF data sets - 100, 101 and 102 records
  - GTF trace files
  - MVS / TSO data sets produced by Collect Report Data
    - ISPF Interface
    - Performance Warehouse in PE Client
  - Data created by the SAVE command

Output

- Traces and Reports
  - Traces show individual DB2 events in order of occurrence
  - Reports show these events summarized by OMPE identifiers, such as primary authorization ID or plan name
- Data sets with formatted data
  - Use for future reports
  - Load into DB2 tables such as the OMPE Performance Warehouse
- Logs that contain information about OMPE processing
  - The Exception Log identifies records that exceed user limits
  - The IFCID Frequency Distribution Log provides counts of input records by IFCID
  - The Job Summary Log provides a summary of events during OMPE execution
  - The DPMLOG Execution Log shows messages issued during OMPE processing

Report Sets

- Statistics
  - A system-wide view of data
  - Traces show delta values
  - Reports are summarized statistics over user defined intervals
- Accounting
  - Thread related traces and reports
  - SMF/GTF record is written when
    - A thread terminates
    - A thread is reused
    - DBAT (DB access thread) becomes inactive
    - Parallel task completes
  - Contains local and distributed DB2 activity associated with a thread
  - Contains DBRM/Package accounting information

Report Sets

- Locking
  - Provides detailed lock information
    - DB2 transaction locks
      - Suspensions / Timeouts / Deadlocks
    - DB2 drain locks and DB2 claims
    - DB2 lock avoidance and related data, like page latches
    - Global locks in a data sharing environment
- SQL Activity
  - Reports based on SQL statement level
  - Based on threads
  - Trace and Reports include
    - Trigger and nesting level information
    - Events and time spent in DDF processing
    - Time spent in signon, autobind processing
    - Time spent in thread creation and termination

Report Sets

- I/O Activity
  - Shows performance of the I/O events for a database and resources
  - Helps identify potential bottlenecks related to I/O activity
  - One summary report and several detail reports for
    - Active Log
    - Archive Log & Bootstrap Data Set
    - Buffer pool
    - EDM pool
    - Cross-Invalidation in a data sharing environment
- System Parameters
  - Provides information about the configuration parameters of a DB2 system

Report Sets

- Audit
  - Tracks DB2 resource access
  - Helps identifying potential security violations
  - Helps monitoring the granting and revoking of privileges
  - Not strictly a performance report
- Record Trace
  - Presents a listing of DB2 trace records in following levels:
    - Summary
    - Short or Long
    - Dump
- Utility Activity
  - Provides information on the utility and bind activity taking place during the processing of a DB2 application

Report Sets

- Explain
  - Provides access path information for a given SQL statement
    - Access type
    - Indexes to be used
    - Order of table access
    - Order of table join
    - The join method chosen
    - Sort information
    - Locking strategy
    - DB2 catalog information

Fundamental Commands

- TRACE
  - Presents individual or logically related DB2 trace data
- FILE
  - Writes trace data to a data set, loadable to the PDB / PWH
- REPORT
  - Presents consolidated data by selected identifiers and time intervals
- REDUCE
  - Consolidates records with certain common characteristics into a single record
- SAVE
  - Writes consolidated data to a VSAM data set
  - After conversion loadable to the PDB / PWH
- RESTORE
  - Loads consolidated data from a VSAM data set

Useful Subcommands

- Intervals and time frames
  - DATEFORMAT (mm/dd/yy, hh:mm:ss.th)
  - FROM and TO start and end date and time
  - INTERVAL to specify summary interval length
  - BOUNDARY to control alignment of intervals
- Report volume and consolidation controls
  - SHORT for overview report
  - LONG for detailed report
  - TOP to find highest values for a specified field
  - INCLUDE and EXCLUDE to specify filters

Sample Statistics Report Job

```
//jobname  JOB CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
//PEV410   EXEC PGM=DB2PM
//STEPLIB  DD  DSN=hlq.RKANMOD,DISP=SHR
//INPUTDD  DD  DSN=input.trace.records,DISP=SHR
//STWORK   DD  DCB=(RECFM=VBS,LRECL=32756,BLKSIZE=6233),
//             SPACE=(CYL,(50,50))
//DPMLOG   DD  SYSOUT=*
//STRPTDD  DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSIN    DD  *
STATISTICS
            REDUCE
                     INTERVAL(60)
            REPORT
                     FROM(01/15/08,00:00:00.00)
                     TO(01/15/08,23:59:59.99)
                     DDNAME(STRPTDD)
                     LAYOUT(SHORT)
                     NOEXCEPTION
EXEC
//
```

Sample Accounting Report Job

```
//jobname  JOB CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID
//PEV410   EXEC PGM=DB2PM
//STEPLIB  DD  DSN=hlq.RKANMOD,DISP=SHR
//INPUTDD  DD  DSN=input.trace.records,DISP=SHR
//DPMLOG   DD  SYSOUT=*
//ACRPTDD  DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSIN    DD  *
ACCOUNTING
            REPORT
                     FROM(01/15/08,00:00:00.00)
                     TO(01/15/08,23:59:59.99)
                     DDNAME(ACRPTDD)
                     LAYOUT(LONG)
                     INCLUDE (PLANNAME(planname)
EXEC
//
```

Rules of Thumb

- EDM Pool
  - LOAD CT SECT FROM DASD / REQ FOR CT SECTIONS
    - Should approach 0% for high volume transactions
  - LOAD DBD FROM DASD / REQUESTS FOR DBD
    - Should approach 0% for high volume transactions
    - Otherwise, 20% is acceptable
  - FREE PG IN FREE CHAIN / PAGES IN EDM POOL
    - EDM Pool too large if > 20%
  - FAILS DUE TO POOL FULL
    - EDM Pool too small

Rules of Thumb

- Buffer Pool
  - DWT
    - Number of times Deferred Write Threshold reached
    - Application dependent, but should be close to 0 if BP is large
  - VDWQT
    - Number of times Vertical Deferred Write Threshold reached
    - Should be 0 - increase BP size or VDWQT level
  - SPTH (90%)
    - Should be 0, increase BP if greater than 5%
  - DMTH (95%)
    - Should be 0, increase BP not
  - BP Hit Ratio
    - Application dependent
    - 80% for random is good
    - 100% for sequential is good

Documentation

- SC18-9983 Reporting User's Guide Version 4.1.0
  - How to use reports for performance monitoring
- SC18-9984 Report Reference Version 4.1.0
  - Report descriptions
- SC18-9985 Report Command Reference Version 4.1.0
  - Report commands and subcommands